Robust and Consistent Estimation of Generators in Credit Risk

Quantitative Finance

ISSN: 1469-7688 (Print) 1469-7696 (Online) Journal homepage: http://www.tandfonline.com/loi/rquf20

Robust and consistent estimation of generators in credit risk

G. dos Reis & G. Smith

To cite this article: G. dos Reis & G. Smith (2017): Robust and consistent estimation of generators in credit risk, Quantitative Finance, DOI: 10.1080/14697688.2017.1383627

To link to this article: https://doi.org/10.1080/14697688.2017.1383627

Published online: 20 Nov 2017.

Quantitative Finance, 2017
https://doi.org/10.1080/14697688.2017.1383627

Robust and consistent estimation of generators in credit risk

G. DOS REIS†‡* and G. SMITH†

†School of Mathematics, Maxwell Institute for Mathematical Sciences, The University of Edinburgh, Edinburgh, UK
‡Centro de Matemática e Aplicações (CMA), FCT, UNL, Lisbon, Portugal

(*Received 13 December 2016; accepted 15 September 2017; published online 20 November 2017*)

Bond rating Transition Probability Matrices (TPMs) are built over a one-year time-frame and for many practical purposes, like the assessment of risk in portfolios or the computation of banking Capital Requirements (e.g. the new IFRS 9 regulation), one needs to compute the TPM and probabilities of default over a smaller time interval. In the context of continuous time Markov chains (CTMC) several deterministic and statistical algorithms have been proposed to estimate the generator matrix. We focus on the Expectation-Maximization (EM) algorithm by Bladt and Sorensen. [*J. R. Stat. Soc. Ser. B (Stat. Method.)*, 2005, **67**, 395–410] for a CTMC with an absorbing state for such estimation. This work's contribution is threefold. Firstly, we provide directly computable closed form expressions for quantities appearing in the EM algorithm and associated information matrix, allowing easy approximation of confidence intervals. Previously, these quantities had to be estimated numerically and considerable computational speedups have been gained. Secondly, we prove convergence to a single set of parameters under very weak conditions (for the TPM problem). Finally, we provide a numerical benchmark of our results against other known algorithms, in particular, on several problems related to credit risk. The EM algorithm we propose, padded with the new formulas (and error criteria), outperforms other known algorithms in several metrics, in particular, with much less overestimation of probabilities of default in higher ratings than other statistical algorithms.

Keywords: Likelihood inference; Credit risk; Transition probability matrices; EM algorithm; Markov Chain Monte Carlo

JEL Classification: C13, C63, G32

1. Introduction

Credit ratings play a key role not just in the calculation of a bank's capital charge (amount of capital a bank must hold) but also are typically a requirement for corporations wishing to issue bonds. There are different agencies which provide firms with a rating and the credit rating the agency gives a company determines in some respect the financial health of the company. Typically ratings are of the form AAA, AA, A, BBB, BB, B, C, D (although it varies between agencies) with, AAA the best (safest), C the worst (riskiest) and D to imply the firm has defaulted. It is also standard for banks to use their own internal ratings system (see Yavin *et al.* 2014). For an overview of the 'science' involved in the rating procedure (see Cantor 2004).

The main object of interest in this work is the so-called annual *Transition Probability Matrix* (TPM), it is a stochastic matrix which shows the migration probabilities of different rated companies within a year. Rating agencies produce these annually. It is possible that such matrices are not initially stochastic due to company mergers or rounding for example. However, they can be renormalized by methods as described in Kreinin and Sidelnikova (2001) and, as argued in Bangia *et al.* (2002), renormalizing non rated companies across all ratings is indeed the industry standard.

The main problem considered here is that a TPM encases transition probabilities over a 1-year time frame and often in practice one needs a 3 month or 10 day transition matrix for which probabilities of default are lower than those in the TPM. Therefore, one wants to accurately estimate the sub-annual matrix given the annual matrix. In the Basel proposals, Basel 3 Supervision (2013, p. 3) a large part of the risk charge will be measured using ES (expected shortfall), which (as shown in Cont *et al.* (2010)) is extremely sensitive to a small shifts in probabilities. Therefore, accurate and consistent estimation is essential in the calculation. Moreover, with the perspective of the *IFTR 9* regulation, one needs to better estimate the related probabilities of default (PD) and Markov chain's generator since for a company whose risk profile changes significantly one needs to assess its risk throughout the bond's lifetime. As PD is given by exponential functions, smaller initial errors will be compounded in a significant way, especially in long termed

*Corresponding author. Email: G.dosReis@ed.ac.uk

bonds. This is compounded by the known fact, corroborated by our numerical experiments, that certain algorithms overestimate the PD. Our methodology yields a way to obtain point-in-time (PIT) estimates of the Probability of Default (PD) from the through-the-cycle (TTC) estimates.

Credit rating models within the Markovian framework are handy both from a theoretical and numerical perspective. Evidence is given in Lando and Skodeberg (2002) that such Markovian structure is not true in practice, nonetheless, within the Markovian structure, efficient implementation of apt Markovian credit risk models and related risk measuring estimations able to deal with massive portfolios is a challenging problem (see Brigo *et al*. 2014, Rutkowski and Tarca 2015). There have been several models that produce non-Markovian effects such as mixing two generators (Frydman and Schuermann 2008) or considering hidden Markov models (Korolkiewicz 2012), see also Long *et al*. (2011). All non-Markovian models require in one way or another access to additional data for accurate calibration. These data are costly, need to be updated over time and many companies opt to deal only with the TPMs. This work focuses on practitioners that do not have access to the data. The issue of rating momentum will be dealt with in forthcoming research.

1.1. The problem at hand

We take the view of a financial agent who wishes to estimate probabilities of default or assess risk in his portfolio due to credit transitions but does not have access to (the expensive) individual credit rating transitions. The agent only has the annual TPM, say $\mathbf{P}(1)$, and uses a continuous time Markov chain (CTMC), say $(\hat{\mathbf{P}}(t))_{t\geq 0}$, with a finite state space to model the changes in rating over time. Under standard conditions the evolution of the CTMC can be written as $\hat{\mathbf{P}}(t) = e^{\mathbf{Q}t}$ where $\mathbf{Q}$ is the generator matrix. The problem is then to estimate $\mathbf{Q}$ given $\mathbf{P}(1)$. This estimation is non-trivial due to the so-called embeddability problem (not reviewed here). It is discussed in great detail by Israel *et al*. (2001) and, for more of the mathematics and many of the existing results on the embeddability problem, we point the reader to Lin (2001).

Several approaches exist to tackle this estimation problem (Kreinin and Sidelnikova 2001, Israel *et al*. 2001, Trück and Özturkmen 2004, Bladt and Sorensen 2005, Inamura 2006, Bladt and Sorensen 2009), either using deterministic algorithms (e.g. diagonal or weighted adjustment, Quasi-optimization of the generator) or statistical ones (Expectation-Maximization (EM), Markov chain Monte-Carlo (MCMC) ones), see Section 3. We focus on the Expectation-Maximization algorithm of Bladt and Sorensen (2005) for CTMCs and allow for an absorbing states.

1.2. Contribution of this work

(i) We provide sufficient conditions to extend the convergence result of Bladt and Sorensen (2005) to individual parameters rather than just convergence of likelihoods. The conditions presented are trivially satisfied in the context of the TPM problem.

(ii) We derive closed form expressions for the entries of the Hessian of the likelihood function used in the EM algorithm. This eliminated several instability issues appearing in other numerical implementations found in the literature and allows for computational speedups (comparatively). Moreover, the result provides a way to estimate the error of the estimation and assess the nature of the stationary point the algorithm has converged to.

(iii) We give a short overview of known methods and implement them with some modifications as to improve their performance. See Sections 3 & 4 for precise meanings: we apply the algorithms to certain credit risk problems and carry out a simulation study to check the impact in the computation of *risk charges*, namely IRC (Incremental Risk Charge) with VaR (Value at Risk), IDR (Incremental Default Risk) with VaR and IRC with ES (Expected Shortfall). We distinguish portfolio types (mixed, investment or speculative); the impact of different types of generators (stable vs unstable); dependence on the sample size and general convergence. We compare probabilities of default as maps of time across different algorithms and find interesting results.

For the study carried out, the implemented EM algorithm is very competitive. It is slightly slower than the deterministic algorithms but much faster than the MCMC algorithms. It embeds statistical properties like robustness that deterministic algorithms cannot capture.

Remark 1.1 We focus purely on continuous over discrete time models. Continuous time algorithms yield robust estimators while the discrete ones do not, with robustness understood in the following sense: from $\mathbf{P}(1)$ estimate $\mathbf{P}(0.5)$ and $\mathbf{P}(0.25)$. From $\mathbf{P}(0.5)$ estimate $\mathbf{P}(0.25)$ again. Continuous algorithms yield the same $\mathbf{P}(0.25)$, discrete algorithms (in general) will not.

Remark 1.2 [Software availability] The findings of this work will appear in the updated version of the CRAN R-package *ctmcd: Estimating the Parameters of a Continuous-Time Markov Chain from Discrete-Time Data* (see Pfeuffer 2017) — https://CRAN.R-project.org/package=ctmcd

This work is organized as follows. In Section 2, we present the EM algorithm and we state our main theoretical findings. In Section 3 we briefly present other known algorithms and in Section 4 we present the benchmarking results.

2. The EM algorithm

There exists extensive literature on the majority of the algorithms we present in this paper, therefore, we only provide brief discussions and include references for additional information. Further, we will use the theory of Markov chains extensively. We do not provide details of the theory, however, interested readers can consult texts such as Norris (1998).

2.1. Preliminaries and standing convention

Throughout this manuscript, we consider companies defined on a finite state space $\{1, \ldots, h\}$, where each state corresponds

to a rating. We denote AAA as rating 1 and D (default) as rating h. We adopt the standard notation that $\mathbf{P}$ is an h-by-h stochastic matrix, which will be the observed TPM (at, say, time $t = 1$) and $\mathbf{Q}$ is an h-by-h generator matrix. We further denote by $P_{ij} := (\mathbf{P})_{ij}$, by $q_{ij} := (\mathbf{Q})_{ij}$ and the intensity of state i by $q_i = \sum_{j \neq i} q_{ij}$ where $i, j \in \{1, \ldots, h\}$. A standard assumption used in credit risk modelling is that default is an absorbing state, hence $P_{hh} = 1$. We work with infinitesimal generators of the following class.

Definition 2.1 [Stable-Conservative infinitesimal Generator matrix of a CTMC] We say a matrix $\mathbf{Q}$ is a generator matrix if the following properties are satisfied for all $i, j \in \{1, \ldots, h\}$: i) $0 \leq q_{ij} < \infty$ for $i \neq j$; ii) $q_{ii} \leq 0$; and iii) $\sum_{j=1}^{h} q_{ij} = 0$ $\forall i$.

If a matrix $\mathbf{Q}$ satisfies the above properties, then for all $t \geq 0$ the matrix $\mathbf{P}(t) := e^{\mathbf{Q}t}$ is a stochastic matrix, where $e^{\mathbf{A}}$ is the matrix exponential of matrix $\mathbf{A}$ (Norris 1998, p. 63). The goal of the algorithms presented is to calculate a generator matrix $\mathbf{Q}$ such that $e^{\mathbf{Q}t}$ is the best fit to the observed TPM, where t denotes the length of time between the rating updates (typically one year).

Throughout let $(X(t))_{t \geq 0}$ denote a CTMC over the finite state space $\{1, \ldots, h\}$ with a generator $\mathbf{Q}$ of the above class. Associated to $X(t)$ is, for i, j in the state space, $K_{ij}(t)$ the number of jumps from i to j in the interval $[0, t]$ and by $S_i(t)$ the holding time of state i in the interval $[0, t]$.

Remark 2.2 If a matrix $\mathbf{P}$ is embeddable,† the algorithms below are pointless and one can easily tackle the problem through eigenvalue decomposition, etc. Or in the case where the exact timing of rating transitions are known one can use the standard maximum likelihood estimator as in Jarrow *et al.* (1997). In our examples the only data given is a set of yearly TPMs which in general are not embeddable and the methods just mentioned do not yield useful results.

2.2. The algorithm

Many methods have been developed in statistics in order to calculate the maximum likelihood estimate, but many methods break in the presence of missing data. Mathematically, we are interested in some set $\mathcal{X}$, but we are only able to observe $\mathcal{Y}$, with the assumption there is a many-to-one mapping from $\mathcal{X}$ to $\mathcal{Y}$. That is, $\mathcal{X}$ is a much richer set than $\mathcal{Y}$. When dealing with such a case, the Expectation Maximization (EM) algorithm often offers a robust solution to the problem. McLachlan and Krishnan (2007) provide a complete overview of the algorithm.

The basis of algorithm is, we observe data y which is a realization (element) of $\mathcal{Y}$. We know y has density function g (sometimes referred to as a sampling density) depending on parameters Ψ in some space Λ, but we want the density (likelihood) of $\mathcal{X}(y)$. Hence, postulate some family of densities f, dependent on Ψ, where f corresponds to the density of the complete data-set $\mathcal{X}(y)$ (the set of points $x \in \mathcal{X}$ which are in the pre-image of $y \in \mathcal{Y}$). The relation between f and g is,

$$g(y; \Psi) = \int_{\mathcal{X}(y)} f(x; \Psi) \mathrm{d}x \,.$$

The idea is, the EM algorithm maximizes g w.r.t. Ψ, but we force it to do so using the density f. Further, define,

$$R(\Psi'; \Psi) := \mathbb{E}_{\Psi}\Big[\ln\big(f(x; \Psi')\big) \Big| y \Big] \qquad \text{for } \Psi', \Psi \in \Lambda, \tag{2.1}$$

where $\mathbb{E}_{\Psi}[\cdot|y]$ is the conditional expectation, conditional on y under parameters Ψ. We assume R to exist for all pairs (Ψ', Ψ), in particular we assume $f(x; \Psi) > 0$ almost everywhere in $\mathcal{X}$ for all Ψ (otherwise the logarithm is infinite). Let us clarify, f is calculated using Ψ', but the expectation is calculated using Ψ. The EM algorithm is then the following iterative procedure.

(i) Choose an initial $\Psi^{(1)}$ and take $p = 1$.
(ii) E-step: Compute $R(\Psi; \Psi^{(p)})$.
(iii) M-step: Choose $\Psi^{(p+1)}$ to be the value of $\Psi \in \Lambda$ that maximizes $R(\Psi; \Psi^{(p)})$.
(iv) Check if the predefined convergence criteria is met, if not, take $p = p + 1$ and return to (ii).

2.2.1. The particular problem of generator estimation. For our problem the observed process is a discrete time Markov chain (DTMC), the unobserved process to estimate is a continuous time Markov chain (CTMC). Therefore, the observed data is the discrete transitions and the parameters we wish to estimate are the entries in the generator. The likelihood of a continuous time fully observed Markov chain with generator $\mathbf{Q}$ is given by the following expression (see Küchler and Sorensen 1997, Chapter 3.4),

$$L_t(\mathbf{Q}) = \exp\left\{ \sum_{i=1}^{h} \left[\sum_{j \neq i} \log(q_{ij}) K_{ij}(t) - S_i(t) \sum_{j \neq i} q_{ij} \right] \right\},$$

with K and S the same as before (immediately before Remark 2.2). Hence, given two generators $\mathbf{Q}', \mathbf{Q}$, the function R in (2.1) is,

$$R(\mathbf{Q}'; \mathbf{Q}) = \sum_{i=1}^{h} \left[\sum_{j \neq i} \log(q'_{ij}) \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y] - \mathbb{E}_{\mathbf{Q}}[S_i(t)|y] \sum_{j \neq i} q'_{ij} \right], \tag{2.2}$$

where y denotes the discrete time observations. Maximizing for q'_{ij} in $R(\mathbf{Q}'; \mathbf{Q})$ yields

$$q'_{ij} = \frac{\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y]}{\mathbb{E}_{\mathbf{Q}}[S_i(t)|y]} \,. \tag{2.3}$$

The difficult step is the calculation of $\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y]$ and $\mathbb{E}_{\mathbf{Q}}[S_i(t)|y]$. We follow an approach similar to Bladt and Sorensen (2005) (see also Bladt *et al.* 2002) but express the result in a framework more suited to the problem of generator estimation from TPMs, rather than the estimation from individual movements. Furthermore, the result derived in Bladt *et al.* (2002) is for irreducible Markov chains making it not applicable to our case (CTMC with absorbing states), accounted for in Proposition 2.4.

†In this setting a stochastic matrix $\mathbf{P}$ is embeddable if there exists a generator $\mathbf{Q}$ such that $\mathbf{P} = e^{\mathbf{Q}}$.

Consider the following functions (see Bladt *et al*. 2002), for $1 \leq i, j \leq h$

$$V_{ij}^*(\mathbf{c}, \mathbf{Z}; t) = \mathbb{E}_{\mathbf{Q}}\left[\exp\left\{ -\sum_{\mu=1}^{h} c_\mu S_\mu(t) \right\} \times \prod_{\mu,\nu=1}^{h} Z_{\mu\nu}^{K_{\mu\nu}(t)} \mathbb{1}_{\{X(t)=j\}} \middle| X(0) = i \right], \quad (2.4)$$

where we denote by $\mathbf{c} = (c_1, \cdots, c_h) \in \mathbb{R}^h$ and $\mathbf{Z} \in \mathbb{R}^{h\times h}$ with $Z_{ii} = 1$ for $i \in \{1, \cdots, h\}$. Observe that V_{ij}^* is the Laplace-Stieltjes transform of the holding times S and the probability generating function of the jumps K, with initial and final states $X(0) = i$ and $X(t) = j$ respectively. Denoting by $\mathbf{V}^*(\mathbf{c}, \mathbf{Z}; t)$ the h-by-h matrix of elements $V_{ij}^*(\mathbf{c}, \mathbf{Z}; t)$. This allows us to give the main theorem (similar version) in Bladt *et al*. (2002).

THEOREM 2.3 *For $t \geq 0$, the matrix $\mathbf{V}^*(\mathbf{c}, \mathbf{Z}; t)$ is given by,*

$$\mathbf{V}^*(\mathbf{c}, \mathbf{Z}; t) = \exp\{[\mathbf{Q} \bullet \mathbf{Z} - \Delta(\mathbf{c})]t\},$$

where $\bullet$ is the Schur (Hadamard) product† of matrices, $\mathbf{Q}$ is the generator matrix from the CTMC and $\Delta(\mathbf{c})$ is the diagonal matrix with entries c_i at position ii for $i = 1, \ldots, h$.

A closed form expression for the expectation terms in (2.3) follows from a result in Van Loan (1978) (sketched also in Hobolth and Jensen (2011)).

PROPOSITION 2.4 *Let $\mathbf{e}_i$ be the column vector of length h which is one at entry i and zero elsewhere, further let us define the $2h$-by-$2h$ matrices $\mathbf{C}_\gamma^{(\alpha\beta)}$ and $\mathbf{C}_\phi^{(\alpha)}$ as,*

$$\mathbf{C}_\gamma^{(\alpha\beta)} = \begin{bmatrix} \mathbf{Q} & q_{\alpha\beta}\mathbf{e}_\alpha\mathbf{e}_\beta^{\mathbf{T}} \\ 0 & \mathbf{Q} \end{bmatrix} \quad \text{and}$$

$$\mathbf{C}_\phi^{(\alpha)} = \begin{bmatrix} \mathbf{Q} & \mathbf{e}_\alpha\mathbf{e}_\alpha^{\mathbf{T}} \\ 0 & \mathbf{Q} \end{bmatrix} \quad \alpha, \beta \in \{1, \cdots, h\}. \quad (2.5)$$

Consider a CTMC X that we observe at n time points $0 \leq t_1 < t_2 < \cdots < t_n$ and denote by y_s the state of the Markov chain at time t_s, i.e. $y_s := X(t_s)$. Then, the expected jumps and holding times over the observations are,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y] = \sum_{s=1}^{n-1} \frac{\left(\exp\{\mathbf{C}_\gamma^{(ij)}(t_{s+1} - t_s)\}\right)_{y_s, h+y_{s+1}}}{(\exp\{\mathbf{Q}(t_{s+1} - t_s)\})_{y_s, y_{s+1}}},$$

$$\mathbb{E}_{\mathbf{Q}}[S_i(t)|y] = \sum_{s=1}^{n-1} \frac{\left(\exp\{\mathbf{C}_\phi^{(i)}(t_{s+1} - t_s)\}\right)_{y_s, h+y_{s+1}}}{(\exp\{\mathbf{Q}(t_{s+1} - t_s)\})_{y_s, y_{s+1}}}.$$

Proof. Observe that $\mathbf{V}^*$ in (2.4) satisfies the differential equation in the statement of Theorem 2.3 (see Bladt and Sorensen 2005). The proof is omitted as one can solve the equation by employing the methods in Van Loan (1978). □

Thus we obtain closed form expressions for the two key quantities appearing in (2.3). This approach differs from Bladt and Sorensen (2005) where they describe numerical schemes to solve the differential equations, namely Runge-Kutta and uniformization. These techniques can yield good results at this level, but our closed form expression will pay dividends when it comes to error estimation.

This yields the relation we desire, however, in our example we have an observed TPM (or sequence of TPMs), $\mathbf{P}$, in the case of equal observation windows, t in the interval $[0, T]$ (although it is trivial to generalize) the expectation can be expressed as,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}] = \sum_{u=1}^{N}\sum_{s=1}^{h}\sum_{r=1}^{h} P_{sr}^u(t) \frac{\left(\exp\{\mathbf{C}_\gamma^{(ij)}t\}\right)_{s,h+r}}{(\exp\{\mathbf{Q}t\})_{s,r}},$$

$$\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}] = \sum_{u=1}^{N}\sum_{s=1}^{h}\sum_{r=1}^{h} P_{sr}^u(t) \frac{\left(\exp\{\mathbf{C}_\phi^{(i)}t\}\right)_{s,h+r}}{(\exp\{\mathbf{Q}t\})_{s,r}}, \quad (2.6)$$

where $N = T/t$ (the number of observations) and $\mathbf{P}^u$ is the TPM of the u-th observation.

Remark 2.5 [The reducible case] Previously, we only had observed transitions, hence they must have a non-zero probability of occurring. Here we can sum s and r over the full range because $\mathbf{P}_{sr}(t)$ acts as an indicator of possible transitions, that is, if $P_{sr}(t) = 0$ we set the s, r component as 0. Clearly, if $P_{sr}(t) > 0$, but $(\exp\{\mathbf{Q}t\})_{sr} = 0$, $\mathbf{Q}$ is misspecified.

2.2.2. Likelihood Convergence of the EM algorithm. In the case of this problem Bladt and Sorensen (2005) provide a proof that the likelihood function converges with one small caveat in order to keep the parameter space compact. Namely, they use the following constrained parameter space, $\mathcal{Q}_\epsilon$, which can be achieved by setting, $\mathcal{Q}_\epsilon = \{\mathbf{Q} \in \mathcal{Q} | \det[\exp(\mathbf{Q})] \geq \epsilon\}$ ($\mathcal{Q}$ is the parameter space from Definition 2.1) for some $\epsilon > 0$. Theorem 4 in Bladt and Sorensen (2005) states that the algorithm will converge to a stationary point of the likelihood or hit the boundary of the parameter space they have induced. It is accepted this is a crude approach to solving the problem and further analysis is needed when $\det[\exp(\mathbf{Q})] = \epsilon$. An alternative approach would be to use a penalized likelihood as discussed in McLachlan and Krishnan (2007, p. 214).

2.2.3. Parameter convergence criteria. The above convergence is sufficient for one to conclude convergence of the likelihood. However, it is not sufficient for convergence of the parameters as one cannot state that the series of iterates $\mathbf{Q}^{(k)}$ converge ($\|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\| \to 0$ as $k \to \infty$). From a theoretical standpoint this may not be as important as convergence of the likelihood itself, nonetheless, it is of key importance for applications. For instance, without convergence of the parameters the risk charge different financial agents obtain from the same data may vary wildly, even under very strict convergence conditions. Before proving convergence we require two important points.

Remark 2.6 With (2.6) in mind we assume that for any $s \neq r$ such that $P_{sr}^u(t) = 0$ for all u, we take the starting point $q_{sr}^{(0)} := (\mathbf{Q}^{(0)})_{sr} = 0$. As discussed in Bladt and Sorensen (2005), any point set to zero will stay at zero for all iterations. Note, we are not changing the problem since these terms will converge to zero under the EM algorithm.

†The Shur product of two $h \times h$ matrices A and B is the $h \times h$ matrix with elements $A_{ij}B_{ij}$.

ASSUMPTION 2.7 *[Element constraint] Similar to Bladt and Sorensen (2005), we will use a manual space constraint to obtain the convergence. Take* $1 > \epsilon > 0$*, such that* $\forall i \neq j$*,* $q_{ij} < 1/\epsilon$*. Moreover, we assume adjacent mixing, namely, for* $i \in \{2, \dots, h-1\}$*,* $q_{i,i\pm1} > \epsilon$ *and* $q_{1,2} > \epsilon$*.*

We denote the space of generator matrices which satisfy this condition as Λ_ϵ*.*

The above assumption ensures non-zero entries in the tri-diagonal band and also only finite entries as one can take ϵ as small as we wish. In the case of TPMs associated to credit ratings, such an assumption is trivially satisfied as one generally has diagonally dominant matrices and companies can always be upgraded or downgraded by one, thus $P^u_{i,i\pm1}$ are typically non-zero. Diagonal dominance is sufficient for the generator to be identifiable and therefore entries do not blow up, we discuss the notion of identifiability in Section 2.3.

Proving the parameters converge is more challenging than the likelihoods, however, Wu (1983) provide a sufficient condition for this to occur, namely a sufficient condition for $\|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\| \to 0$ as $k \to \infty$ is, there exists a forcing function† σ such that,

$$R(\mathbf{Q}^{(k+1)}; \mathbf{Q}^{(k)}) - R(\mathbf{Q}^{(k)}; \mathbf{Q}^{(k)}) \geq \sigma(||\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}||)\,.$$

An example of a forcing function is $\sigma(t) = \lambda t^2$ where $\lambda > 0$. We require the following bounds on the expected values to show convergence.

LEMMA 2.8 *Let* N *and* $\mathbf{P}^u$ *be as defined in (2.6) and assume for* $i \neq j$ *there exists a* $u \in \{1, \dots, N\}$ *such that* $P^u_{ij} > 0$ *(we observe a movement from* i *to* j *in observation window* u*). Then we obtain the following bounds on the expected number of jumps:*

$$\begin{aligned} P^u_{ij}\frac{\epsilon q_{ij}}{h} &\leq \mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}] \\ &\leq h^2 N \frac{ht}{\epsilon \min\left\{\epsilon^h t^h \exp\{-th^2/\epsilon\}\,,\ \exp\{ht/\epsilon\}\right\}}\,. \end{aligned} \tag{2.7}$$

Moreover, assuming there exists a $u \in \{1, \dots, N\}$ *such that* $P^u_{ii} > 0$*, we obtain the following bound on the expected holding time,*

$$\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}] \geq P^u_{ii} t \exp\Big\{-\frac{ht}{\epsilon}\Big\}\,. \tag{2.8}$$

To maintain the flow of the text we state immediately our main convergence result, and defer the proof of the Lemma to Appendix A.1.

THEOREM 2.9 *Under Assumption 2.7, then, there exists a* $\lambda > 0$ *such that for all EM iterations* $k \in \mathbb{N}$*,*

$$R\big(\mathbf{Q}^{(k+1)}; \mathbf{Q}^{(k)}\big) - R\big(\mathbf{Q}^{(k)}; \mathbf{Q}^{(k)}\big) \geq \lambda \|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\|^2\,,$$

where $||\cdot||$ *is the Euclidean norm.*

†A forcing function is defined as any function $\sigma : [0,\infty) \to [0,\infty)$ such that for any sequence t_k defined in $[0,\infty)$, $\lim_{k\to\infty}\sigma(t_k) = 0$ implies $\lim_{k\to\infty} t_k = 0$.

Proof. Writing out the difference in the R terms we obtain,

$$\sum_{i=1}^{h}\sum_{j\neq i}\Big[\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(t)|\mathbf{P}]\big(\log(q_{ij}^{(k+1)}) - \log(q_{ij}^{(k)})\big) - \mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}]\big(q_{ij}^{(k+1)} - q_{ij}^{(k)}\big)\Big]\,.$$

Due to the form of the Euclidean norm squared and the function R, it is sufficient to show the inequality holds for all $i \neq j$. Namely, it is sufficient to show the existence of a $\lambda > 0$ such that,

$$\begin{aligned} &\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]\Big(\log(q_{ij}^{(k+1)}) - \log(q_{ij}^{(k)})\Big) \\ &\quad - \mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}]\left(q_{ij}^{(k+1)} - q_{ij}^{(k)}\right) \geq \lambda\left(q_{ij}^{(k+1)} - q_{ij}^{(k)}\right)^2, \end{aligned} \tag{2.9}$$

for all $i \neq j$. We tackle the log terms first. It is well-known that we can express any C^∞-function using Taylor expansion to a finite number of terms with some error (remainder) term. Moreover, the error term has a known form and hence, using an exact Taylor expansion to second order, there exists a $Z \in [\min(q_{ij}^{(k)}, q_{ij}^{(k+1)}), \max(q_{ij}^{(k)}, q_{ij}^{(k+1)})]$ such that,

$$\begin{aligned} &\log\left(q_{ij}^{(k+1)}\right) - \log\left(q_{ij}^{(k)}\right) \\ &\quad = \frac{-1}{q_{ij}^{(k+1)}}\left(q_{ij}^{(k)} - q_{ij}^{(k+1)}\right) + \frac{1}{2Z^2}\left(q_{ij}^{(k)} - q_{ij}^{(k+1)}\right)^2, \end{aligned}$$

where we have expanded $q_{ij}^{(k)}$ around $q_{ij}^{(k+1)}$. Substituting (2.3) into the LHS of (2.9), the condition simplifies to,

$$\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \geq \lambda(q_{ij}^{(k+1)} - q_{ij}^{(k)})^2\,.$$

In order to show this bound we need to get a handle on Z. Clearly, there are two options between iterations, either $q_{ij}^{(k)} > q_{ij}^{(k+1)}$ or $q_{ij}^{(k)} \leq q_{ij}^{(k+1)}$. In the latter case we obtain,

$$\begin{aligned} &\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \\ &\quad \geq \frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}]^2}{2\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2\,. \end{aligned}$$

Since we can element wise bound $\mathbf{Q}^{(k)}$, using Lemma 2.8 and Assumption 2.7 we can bound the term $\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]$ from above and $\mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}]$ from below by constants (depending on ϵ). Hence, we can choose a λ independent of k such that the condition is satisfied.

The second case $q_{ij}^{(k)} > q_{ij}^{(k+1)}$, follows a similar argument. Again, we can set Z as the larger of the two values, thus we obtain the following inequality,

$$\begin{aligned} &\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \\ &\quad \geq \frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2\left(q_{ij}^{(k)}\right)^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2\,. \end{aligned}$$

Using Lemma 2.8, we can reduce this inequality to,

$$\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \geq \frac{P^u_{ij}\epsilon}{2hq_{ij}^{(k)}}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2\,.$$

Since $P^u_{ij} > 0$ and we can bound each q_{ij} from above, again we choose a λ independent of k. □

2.2.4. Starting value for the EM algorithm. The final point to discuss, is the choice of the initial matrix $\mathbf{Q}$. It is useful from a computational point of view to start in a good place. Here we choose $\mathbf{Q}$ based on a generalization of the QOG algorithm (described in Section 3.1) that allows for complex inputs. For each entry q_{ij} we define the input as,

$$q_{ij} \to \operatorname{sign}\big(\operatorname{Re}(q_{ij})\big) \times |q_{ij}|\,,$$

where $|q_{ij}|$ is the magnitude of q_{ij} and $\operatorname{Re}(q_{ij})$, is the real component of q_{ij}. With the newly defined $\mathbf{Q}$ we apply the QOG algorithm. We take any zero entries not in the final row to be a small number (10^{-5}, say) unless there are zero observed transitions. This defines our initial choice of $\mathbf{Q}$. We define the EM algorithm steps as,

(i) Take an initial intensity matrix $\mathbf{Q}$ and positive value ϵ.
(ii) While the convergence criteria is not met and all entries of $\mathbf{Q}$ are within the boundaries
  (1) E-step: calculate $\mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}]$ and $\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}]$.
  (2) M-step: set $q'_{ij} = \mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}]/\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}]$, for all $i \neq j$ and set q_{ii} appropriately.
  (3) Set $\mathbf{Q} = \mathbf{Q}'$ (where $\mathbf{Q}'$ is the matrix of q's) and return to E-step.
(iii) End while and return $\mathbf{Q}$.

This leads to the following theorem for convergence of the EM.

THEOREM 2.10 *[Convergence of the EM] Assume that our initial point is in the parameter space Λ_ϵ: is a true generator and satisfies Assumption 2.7. Then either the sequence of points $\{\mathbf{Q}^{(k)}\}_k$ converges to a single point in Λ_ϵ which is also a stationary point of the likelihood, or the entries go to the boundary (blow up or some tri-diagonal elements in an non-absorbing row go to zero).*

A proof of Theorem 2.10 follows directly from Theorem 4 in Bladt and Sorensen (2005) and our Theorem 2.9.

Remark 2.11 [The unique maximizer of the Likelihood] The natural question one may ask is does this stronger form of convergence imply convergence to the global maximum? The problem of existence and uniqueness of maximum likelihoods in this setting is a very challenging problem with a long history. Bladt and Sorensen (2005) give a wonderful overview on the subject, Theorem 1 in Bladt and Sorensen (2005) also provides results on existence and uniqueness of the maximum. Unfortunately, one cannot say more than this, if one can derive conditions under which a unique maximum existed (for non-embeddable TPMs) then the above convergence result is sufficient to conclude the EM will converge to the MLE.

During the final revision of this manuscript, Pfeuffer *et al.* (2017) came to our attention. There, the authors propose two algorithms to mitigate the effect of the EM's possible convergence towards a local but not necessarily the global maximum of the likelihood function (no proofs are given). Our Theorem 2.9 is handy in this context as it shows that once the EM lands 'near' the global maximum the iteration will converge to it.

2.3. Variance estimation

In this section, we derive an expression for the Hessian of the likelihood. We use a result in Oakes (1999) and follow Bladt and Sorensen (2009), however, unlike Bladt and Sorensen (2009), we provide a closed form expression for the Hessian. This result eliminates the stability problems observed in the numerical simulation case when the entries in $\mathbf{Q}$ are small. The Hessian provides a way to estimate the standard error of the maximum likelihood estimates and further allows us to assess the nature of the converged stationary point (this is further discussed in Section 4.4.1).

We point the reader to Bladt and Sorensen (2005, Theorem 1) for results on the existence and uniqueness of maximum likelihood estimators with respect to this problem. Further, for discussions on consistency and asymptotic normality related to this problem one should consult (Kremer and Weißbach 2013, Kremer and Weißbach 2014). Kremer and Weißbach (2013), provide sufficient conditions for consistency, the key assumption relies on so-called model *identifiability*.† Kremer and Weißbach (2013) prove *identifiability* under conditions which are too restrictive for our purpose; (Bladt and Sorensen 2005, Dehay and Yao 2007) discusses the problem of *identifiability* in detail. From Cuthbert (1973); Bladt and Sorensen (2005) for the model to be identifiable it is sufficient (though very crude) to have $\min_i(\exp\{\mathbf{Q}t\})_{ii} > 1/2$, Culver (1966) gives a requirement for general matrices based on the eigenvalues which one can always aposteriori verify after a Q is deduced. The crucial assumption in Kremer and Weißbach (2014) to obtain asymptotic normality, is that the Hessian must be invertible at the true value, we can of course verify invertibility a posteriori.

We recall a result from Oakes (1999) for calculating the Hessian of the likelihood.

LEMMA 2.12 *The second derivative of the likelihood with parameter Ψ and observed information y is related to the EM function R by*

$$\frac{\partial^2 L(\Psi; y)}{\partial \Psi^2} = \left[\frac{\partial^2 R(\Psi'; \Psi)}{\partial \Psi'^2} + \frac{\partial^2 R(\Psi'; \Psi)}{\partial \Psi' \partial \Psi}\right]_{\Psi'=\Psi}.$$

Injecting (2.2) in the above we obtain,

$$\frac{\partial^2 R(\mathbf{Q}'; \mathbf{Q})}{\partial q'_{\alpha\beta}\partial q'_{\mu\nu}} = \frac{-1}{q'^2_{\mu\nu}}\mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y]\delta_{\alpha\mu}\delta_{\beta\nu}\,, \tag{2.10}$$

$$\frac{\partial^2 R(\mathbf{Q}'; \mathbf{Q})}{\partial q_{\alpha\beta}\partial q'_{\mu\nu}} = \frac{1}{q'_{\mu\nu}}\frac{\partial}{\partial q_{\alpha\beta}}\mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y] - \frac{\partial}{\partial q_{\alpha\beta}}\mathbb{E}_{\mathbf{Q}}[S_\mu(t)|y]\,, \tag{2.11}$$

where δ_{ab} is the Kronecker delta. From our previous work, (2.10) is easy to obtain, however, (2.11) involves derivatives of the expected jumps and holding times and is thus challenging. Bladt and Sorensen (2009) opt for a simple numerical scheme to compute these derivatives and found unstable results, although the authors do remark that more sophisticated numerical schemes could yield improved results at greater computational expense.

†In our setting a model is identifiable if there are no two generators $\mathbf{Q} \neq \mathbf{Q}'$ such that $\exp\{\mathbf{Q}t\} = \exp\{\mathbf{Q}'t\}$.

It is worth noting we have made no comment on the allowed values of α, β, μ and ν, other than the clear restriction that they belong to $\{1, \ldots, h\}$. Let us now state the following definition.

Definition 2.13 [Allowed pairs] We say that the pair α, β is allowed if $\alpha \neq \beta$ (not in the diagonal) and $q_{\alpha\beta}$ is not converging to zero under the EM algorithm.

For practical applications, one can imagine the set of allowed values, as the set of α, β such that $q_{\alpha\beta} > \epsilon$, where ϵ is some cut-off point (10^{-8}, say). The reason we must exclude small parameters is, this analysis only holds in the large data limit, since we do not have an infinite amount of data we cannot for certain rule out some jump, however, if $q_{\alpha\beta}$ is converging to zero, it implies that this parameter is either zero or extremely close to zero, and therefore, we can bound it above by a small number. Moreover, from a mathematical point of view a parameter which does tend to zero (or even becomes zero) lies on the boundary, where the notion of differentiability is not clear. Therefore, we can think of the 'allowed pairs' as the variables when solving the problem in a restricted parameter space. We now present the following theorem.

THEOREM 2.14 *Let $\mu, \nu, \alpha, \beta \in \{1, \ldots, h\}$, and $\mathbf{Q}$, $\mathbf{Q}'$ be two generator matrices ($\in \Lambda_\epsilon$ for some $\epsilon > 0$). For any two allowed pairs α, β and μ, ν, the derivative in (2.11) is,*

$$\begin{aligned}
&\frac{\partial^2 R(\mathbf{Q}'; \mathbf{Q})}{\partial q_{\alpha\beta} \partial q'_{\mu\nu}} \\
&= \sum_{s=1}^{n-1} \frac{1}{q'_{\mu\nu}} \Bigg[-(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s, y_{s+1}} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} \\
&\qquad \times (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} + (e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s, y_{s+1}} \\
&\qquad \times \left(e^{\mathbf{C}_\psi^{(\alpha\beta,\mu\nu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \Bigg] \\
&\quad - \Bigg[-(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s, y_{s+1}} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} \\
&\qquad \times (e^{\mathbf{C}_\phi^{(\mu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} + (e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s, y_{s+1}} \\
&\qquad \times \left(e^{\mathbf{C}_\omega^{(\alpha\beta,\mu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \Bigg],
\end{aligned}$$

where the $2h$-by-$2h$ matrices, $\mathbf{C}_\gamma^{(\alpha\beta)}$, $\mathbf{C}_\phi^{(\alpha)}$, $\mathbf{C}_\eta^{(\alpha\beta)}$, are defined as,

$$\mathbf{C}_\gamma^{(\alpha\beta)} = \begin{bmatrix} \mathbf{Q} & q_{\alpha\beta}\mathbf{e}_\alpha\mathbf{e}_\beta^{\mathcal{T}} \\ 0 & \mathbf{Q} \end{bmatrix}, \mathbf{C}_\phi^{(\alpha)} = \begin{bmatrix} \mathbf{Q} & \mathbf{e}_\alpha\mathbf{e}_\alpha^{\mathcal{T}} \\ 0 & \mathbf{Q} \end{bmatrix},$$
$$\mathbf{C}_\eta^{(\alpha\beta)} = \begin{bmatrix} \mathbf{Q} & \mathbf{e}_\alpha\mathbf{e}_\beta^{\mathcal{T}} - \mathbf{e}_\alpha\mathbf{e}_\alpha^{\mathcal{T}} \\ 0 & \mathbf{Q} \end{bmatrix},$$

and the $4h$-by-$4h$ matrices $\mathbf{C}_\psi^{(\alpha\beta,\mu\nu)}$, $\mathbf{C}_\omega^{(\alpha\beta,\mu)}$ are defined

$$\mathbf{C}_\psi^{(\alpha\beta,\mu\nu)} = \begin{bmatrix} \mathbf{C}_\gamma^{(\mu\nu)} & \frac{\partial \mathbf{C}_\gamma^{(\mu\nu)}}{\partial q_{\alpha\beta}} \\ 0 & \mathbf{C}_\gamma^{(\mu\nu)} \end{bmatrix}, \mathbf{C}_\omega^{(\alpha\beta,\mu)} = \begin{bmatrix} \mathbf{C}_\phi^{(\mu)} & \frac{\partial \mathbf{C}_\phi^{(\mu)}}{\partial q_{\alpha\beta}} \\ 0 & \mathbf{C}_\phi^{(\mu)} \end{bmatrix}.$$

The proof of this uses similar techniques to Proposition 2.4 along with differentiation properties of matrix-exponentials, and is deferred to Appendix A.2.

Remark 2.15 In the above representation for the derivative of R, we use subscripts of the form $h + y_{s+1}$ and $3h + y_{s+1}$, this is simply a consequence of the result in Van Loan (1978). Namely, we are not interested in all the entries of the matrix, only an h-by-h segment. We therefore need to adjust the indexing to only take elements at this specific segment.

Using Theorem 2.14 and Lemma 2.12, we can write the elements of the Hessian corresponding to the $q_{\alpha\beta}q_{\mu\nu}$ derivative as,

$$\begin{aligned}
&\frac{\partial^2 L(\mathbf{Q}; y)}{\partial q_{\alpha\beta} \partial q_{\mu\nu}} \\
&= \sum_{s=1}^{n-1} \frac{-1}{q_{\mu\nu}^2} (e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s, y_{s+1}} (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \delta_{\alpha\mu}\delta_{\beta\nu} \\
&\quad + \frac{1}{q_{\mu\nu}} \Bigg[-(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s, y_{s+1}} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} \\
&\qquad \times (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} + (e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s, y_{s+1}} \\
&\qquad \times \left(e^{\mathbf{C}_\psi^{(\alpha\beta,\mu\nu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \Bigg] \\
&\quad - \Bigg[-(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s, y_{s+1}} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} \\
&\qquad \times (e^{\mathbf{C}_\phi^{(\mu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} + (e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s, y_{s+1}} \\
&\qquad \times \left(e^{\mathbf{C}_\omega^{(\alpha\beta,\mu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \Bigg].
\end{aligned}$$

A similar transform to (2.6) can be applied here to obtain the Hessian from TPMs. When using the result to estimate the error, some knowledge of the number of companies per rating is required.

2.3.1. Computation of the error. Due to the Hessian only being defined for parameters $q_{\alpha\beta} > 0$, some parameters are not included in the Hessian, thus the matrix is smaller than $(h-1)^2$-*by*-$(h-1)^2$. We compute the Hessian as follows,

- Let N_a be the number of allowed points in the estimated $\mathbf{Q}$ returned from the EM algorithm.
- Define an N_a-*by*-2 matrix $\mathbf{V_Q}$ as the matrix which records the allowed (in the sense of Definition 2.13) components of $\mathbf{Q}$. The ijth component of the Hessian is then the differential,
$$\frac{\partial^2}{\partial q_{\mathbf{V_Q}(i,1)\mathbf{V_Q}(i,2)} \partial q_{\mathbf{V_Q}(j,1)\mathbf{V_Q}(j,2)}}.$$
- If we denote the Hessian by the N_a-*by*-N_a matrix $\mathbf{H}(\cdot)$, then the information matrix is given by $-\mathbf{H}(\cdot)$. The estimated variance of the allowed parameter q_{ab} is then the ith diagonal element of $-\mathbf{H}(\cdot)^{-1}$, where $\mathbf{V_Q}(i, 1) = a$ and $\mathbf{V_Q}(i, 2) = b$.
- Following standard statistics, the normal based 95% confidence interval for q_{ab} is $q_{ab} \pm 1.96\sqrt{Var(q_{ab})}$.

3. Competitor algorithms

3.1. Deterministic algorithms

3.1.1. Diagonal adjustment (DA). The first method to discuss is diagonal adjustment, see Israel *et al.* (2001). Given a TPM, $\mathbf{P}$, one calculates the matrix logarithm directly. However,

due to the embeddability problem, the logarithm may not be a valid generator. To solve this problem Israel *et al*. (2001) suggest setting for $i \neq j$,

$$q_{ij}^{DA} = \begin{cases} (\log(\mathbf{P}))_{ij}\,, & \text{if } (\log(\mathbf{P}))_{ij} \geq 0\,, \\ 0\,, & \text{otherwise.} \end{cases}$$

and adjusting (re-balancing) the diagonal element correspondingly, $q_{ii}^{DA} = \sum_{j\neq i} -q_{ij}$ for $i \in \{1, \cdots, h\}$.

Hence forcing the corresponding matrix $\mathbf{Q}^{DA}$ to satisfy the properties of a generator.

3.1.2. Weighted Adjustment (WA). Weighted adjustment is also suggested in Israel *et al*. (2001). It follows diagonal adjustment except, one re-balances across the entire row. Again, calculate the logarithm of the TPM to find q's, then compute

$$G_i = |q_{ii}| + \sum_{j\neq i} \max(q_{ij}, 0)\,, \qquad B_i = \sum_{j\neq i} \max(-q_{ij}, 0)\,.$$

The entries corresponding to weighted adjustment are defined as,

$$q_{ij}^{WA} = \begin{cases} 0 & \text{if } i \neq j \text{ and } q_{ij} < 0\,, \\ q_{ij} - B_i|q_{ij}|/G_i & \text{otherwise if } G_i > 0\,, \\ q_{ij} & \text{otherwise if } G_i = 0\,. \end{cases}$$

3.1.3. Quasi-Optimization of the Generator (QOG). The above two methods are unfortunately not optimal in any sense. The QOG (Quasi-Optimization of the Generator), method suggested in Kreinin and Sidelnikova (2001) relies on optimization and is therefore an improvement on the diagonal and weighted adjustment methods. QOG involve solves the minimization problem $\min_{\mathbf{Q}\in\mathcal{Q}} \|\mathbf{Q} - \log(\mathbf{P})\|$, where $\mathcal{Q}$ is the space of stable generator matrices and $||\cdot||$ is the Euclidean norm. Further, Kreinin and Sidelnikova (2001) provide an efficient algorithm to obtain $\mathbf{Q}$.

3.2. Statistical algorithm: Markov Chain Monte Carlo

An alternative statistical algorithm one can adopt is MCMC (Markov Chain Monte Carlo). For the reader's convenience we have included a summary of MCMC in Appendix 2. It should be noted that all MCMC algorithms presented here use a so-called auxiliary variable technique, by introducing the fully observed Markov chain, X as a random variable. Moreover, the prior for $\mathbf{Q}$ is $\Gamma(\alpha, 1/\beta)$ (shape and scale), which is conjugate for the likelihood of a CTMC.

3.2.1. Gibbs sampling - Bladt & Sorensen 2005. To simulate the Markov process, X, Bladt and Sorensen (2005) suggest a rejection sampling method. As is stated in Bladt and Sorensen (2005), such a sampling method runs into difficulties when considering low probability events since the rejection rate will be high (e.g. default for high rated bonds). The MCMC algorithm is summarized as follows, Inamura (2006),

(i) Construct an initial generator $\mathbf{Q}$ using the prior distribution $(\Gamma(\alpha_{ij}, 1/\beta_i))$.

(ii) For some specified number of runs

(1) Simulate X for each observation from Y, with law according to $\mathbf{Q}$.
(2) Calculate the quantities of interest K and S, from X.
(3) Construct a new $\mathbf{Q}$ by drawing samples from $\Gamma(K_{ij}(t) + \alpha_{ij}, 1/(S_i(t) + \beta_i))$.
(4) Save this $\mathbf{Q}$ and use it in the next simulation.

(iii) From the list of $\mathbf{Q}$s, drop some proportion (burn in), then take the mean of the remainder.

The issues with this method are the choice of α and β and the number of runs required before we know that the sample has converged (burn in). Both of these are critical in obtaining accurate answers from MCMC and although Bladt and Sorensen (2005) suggested taking α_{ij} and β_i to be 1, they observe MCMC overestimating entries in the generator when true entries were small. Furthermore, here we are required to use the TPM indirectly through inferring company transitions. That is, we consider M companies in each rating and define the number of companies to make the transition i to j as $M \times P_{ij}$, this of course need not be an integer, but we can always normalize the entries. The reason we cannot use the TPM directly as we did in the EM algorithm is due to the fact that MCMC becomes very sensitive to the values in the prior. The burn in for MCMC will be of little concern to us here as will become apparent when carrying out analysis on the algorithms.

3.2.2. Importance sampling - Bladt & Sorensen 2009. Bladt and Sorensen (2009) address some of the issues in Bladt and Sorensen (2005) by running the same algorithm as previous combined with an importance sampling scheme based on the Metropolis-Hastings algorithm (in its essence a single component Metropolis-Hastings algorithm). The proposal distribution suggested is a Markov chain with generator given by the 'neutral matrix' $\mathbf{Q}^*$, which takes the following form,

$$\mathbf{Q}^* = \frac{1}{W}\left(\mathbf{1}_h - \mathbf{I}_h - h\mathbf{I}_h\right),$$

where $\mathbf{1}_h$ and $\mathbf{I}_h$ is the h-by-h matrix of ones and identity matrix, respectively, and W is a scaling factor set to match the intensities in the true generator matrix $\mathbf{Q}$. Bladt and Sorensen (2009) note that if entries in $\mathbf{Q}$ are known to be zero, then the corresponding element in $\mathbf{Q}^*$ should also be set to zero and the diagonal modified accordingly. Thus transitions rarely produced by the generated Markov chain will occur much more frequently under $\mathbf{Q}^*$. Thus we have solved (at least partially) one of the problems faced in MCMC. The importance sampling weights for a chain X are,

$$w(X) = \frac{L(\mathbf{Q}; X)}{L(\mathbf{Q}^*; X)}\,,$$

where L is the CTMC likelihood. For the priors, Bladt and Sorensen (2009) do not suggest any significant improvement on their earlier work. The authors use $\alpha = 1$ and $\beta = 5$, which they claim gives better results than the suggestion in Bladt and Sorensen (2005). However, it still provides a problem when dealing with entries in $\mathbf{Q}$ which are close to zero. The problem stems from the fact that very little information is known (rarely

observed) for certain transitions, therefore, the output for these entries is mostly based on our prior beliefs.

3.2.3. MCMC mode algorithm. Inamura (2006) presented an alternative algorithm to the original MCMC algorithm presented in Bladt and Sorensen (2005), whereby one calculates the mode rather than the mean. The author claims that this gives extremely accurate results and outperforms other algorithms. The reasoning presented is that the standard MCMC overestimates in the small probability cases due to the gamma distribution being 'skewed', therefore the mode is a better estimate. Inamura (2006) approximates the mode of $\{q_{ij}^{(k)}\}$ by kernel smoothing over the estimates (after taking the log transform to ensure all results are positive).

Remark 3.1 [Other MCMC-based estimators] Many extensions and different MCMC methods to solve this problem are possible (e.g. priors as hyperparameters or sequential Monte Carlo techniques). Here, we consider less complex MCMC algorithms which already set the tone for a comparative study.

4. Benchmarking the algorithms

Due to the diversity of investments bank's make, one cannot assess an algorithms' performance with a single test. With this in mind we consider a host of tests on different portfolios and matrices. The computations were carried out on a Dell PowerEdge R430 with four Intel Xeon E5-2680 processors. During the review process of our work we found Pfeuffer (2017) with an R-implementation of some of the algorithms covered in the previous section. The performance tests of Pfeuffer (2017) are a just subset of those we present next and independently confirm (where there is overlap) our findings, in particular the timing of the MCMC algorithms versus the EM. A version of our algorithms will appear in the mentioned R-package (see Remark 1.2).

The first observation we make is, transition matrices can vary substantially depending on the financial climate (see Christensen *et al*. 2004 and Cantor 2004). Therefore, we consider two different generator matrices which can be thought of as the generator in financial stress and the generator in financial calm. In order to keep these matrices 'reasonable' we start off with the generator given in Christensen *et al*. (2004) built using a large amount of data (see also Inamura (2006)) and consider a generator which has in general higher transition rates and one with lower transition rates. Through considering more than one generator this provides a more detailed assessment of the performance of the various algorithms than other comparative reviews, such as Inamura (2006). The generators we consider are shown in tables 1 and 2. We observe that table 1, has more non-zero entries and larger entries than that of table 2.

Throughout the analysis we refer to the multiple MCMC algorithms introduced in Section 3 which we label in the following way: `MCMC BS05` is Bladt and Sorensen (2005)'s algorithm of Section 3.2.1; `MCMC BS09` is Bladt and Sorensen (2009)'s algorithm of Section 3.2.2; and `MCMC Mode` is Inamura (2006)'s algorithm in Section 3.2.3.

4.1. Sample size inference

The first test we consider is an extension to a test in Inamura (2006), where the author considers a true underlying generator and masks it using it to simulate TPMs, which we view as observations, then applying the algorithms to each observation. The key point here is, Inamura (2006) only simulates 100 companies per rating and hence the outputted TPM is non-embeddable (has 0 entries for accessible jumps). This is an extremely useful test because it provides a fair and intuitive way to assess the performance of each algorithm, however, Inamura (2006) only considers one true generator and only one level of information i.e. 100 companies per rating. Alongside the two different generators we also consider a range of companies per rating to determine its effect on convergence for each algorithm. Furthermore, Inamura (2006) uses seven years worth of data, although one would likely have access to multiple years worth of TPM data, it is unlikely that we would have seven years of transitions from the same generator. Hence we consider four years, which is more consistent with time homogeneity estimates for generators (see Christensen *et al*. 2004). We calculate our estimates for the generator as follows.

(i) Take a range of obligors per rating, [100, 200, 300, 500, 750, 1000] and 10 random seeds.
(ii) For each true generator simulate four one year TPMs for each seed and for each obligor per rating. Hence we have (#Years × #Obligors categories × #Random Seeds × #True generators), simulated TPMs.
(iii) For each set of four simulated TPM we estimate the generator for each algorithm. MCMC may take a long time to run, therefore we consider the time taken to carry out the first 10 runs and the total time taken, if these exceed 180 or 18 000 seconds, respectively, the algorithm is deemed to be too slow and no result is returned. Note, MCMC algorithms use 3000 runs with a burn in of 300. This is smaller than Inamura (2006), for example, however, Inamura (2006) shows apparent convergence to the stationary distribution in a small number of iterations and we observe a similar result.
(iv) Therefore, for each algorithm we have (# Obligors categories × # Random Seeds × # True generators) estimated generators to analyse.

We analyse the estimated generators by considering, distance between estimated generator and true generator in Euclidean norm and difference in one year probability of default. All results presented have been obtained by analysing the estimated generator for each seed, then averaging. This gives a better picture of the average performance.

4.1.1. Convergence in euclidean norm. Our goal in this analysis is to consider the empirical rate of improvement of each algorithm as our 'information' about the true generator increases. For each obligor category we calculate the natural log of the distance (measured by the Euclidean norm) between the estimate and the true. The results are shown in figures 1 and 2.

Note the x-axis is on a logarithmic scale. We observe similarities between the two figures, most notably in the case of low information all algorithms have very similar convergence results,

Table 1. True unstable generator.

	AAA	AA	A	BBB	BB	B	C	D
AAA	–0.146371	0.085881	0.04549	0.015	0	0	0	0
AA	0.018506	–0.166337	0.114831	0.033	0	0	0	0
A	0.0276	0.047012	–0.198043	0.09043	0.023001	0.01	0	0
BBB	0.011469	0.010734	0.088133	–0.243046	0.077569	0.044407	0.010734	0
BB	0	0	0.019159	0.184699	–0.323077	0.106166	0.013053	0
B	0	0	0.012280	0.034822	0.093489	–0.296265	0.134273	0.022401
C	0	0	0	0	0.02	0.140209	–0.600939	0.440730
D	0	0	0	0	0	0	0	0

Table 2. True stable generator.

	AAA	AA	A	BBB	BB	B	C	D
AAA	–0.061371	0.055881	0.005490	0	0	0	0	0
AA	0.013506	–0.096337	0.074831	0.008	0	0	0	0
A	0	0.037012	–0.097442	0.06043	0	0	0	0
BBB	0	0.000734	0.058133	–0.120843	0.057569	0.004407	0	0
BB	0	0	0.009159	0.104699	–0.190024	0.076166	0	0
B	0	0	0	0.024822	0.083489	–0.174985	0.064273	0.002401
C	0	0	0	0	0	0.080209	–0.300939	0.220730
D	0	0	0	0	0	0	0	0

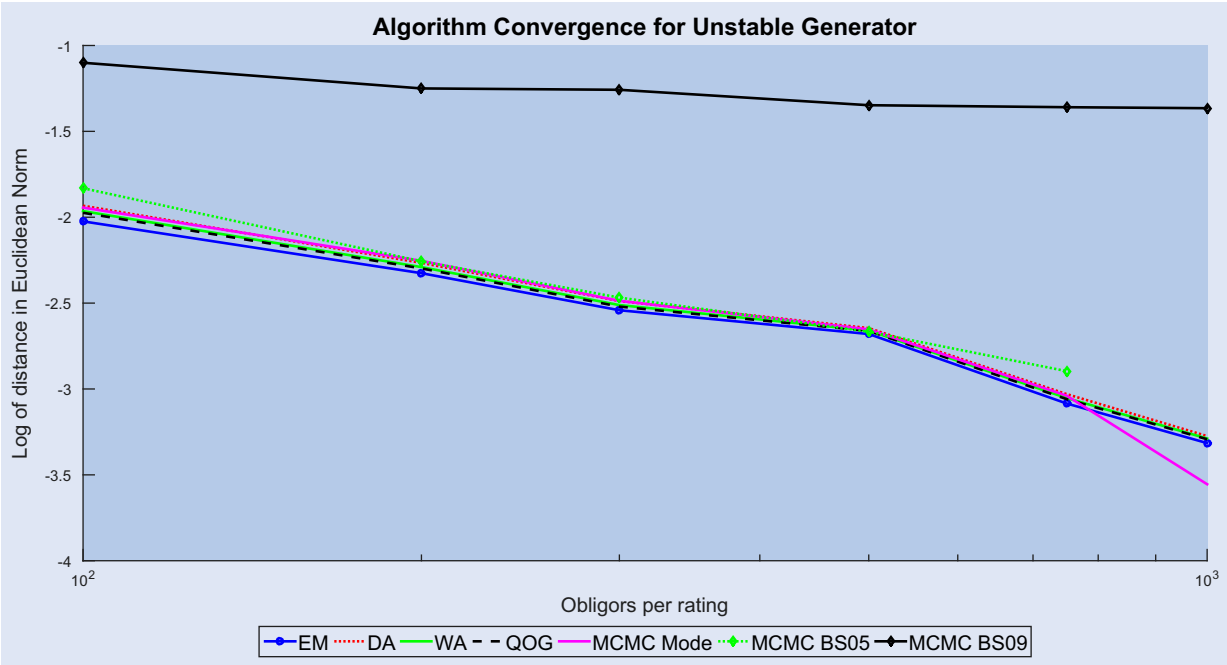

Figure 1. Showing the log of the error for each algorithm as a function of obligors per rating.

however, as we increase the information there is substantial variation in improvement, `MCMC BS09` algorithm does not improve as well as the other algorithms. Missing points stem from an algorithm failing the acceptance times.

The MCMC algorithms have a potentially increased error due to the Monte Carlo simulation, lowering it requires a larger computational expense to the already most expensive algorithm being tested here. For the Bladt and Sorensen (2009) algorithm, the neutral matrix approximation may give poor mixing, thus the additional error.

4.1.2. Error in probability of default. Although overall error is important, it does not provide details on the small probability scale. This is extremely important in banking, since estimation of the probability of default is crucial. Using the

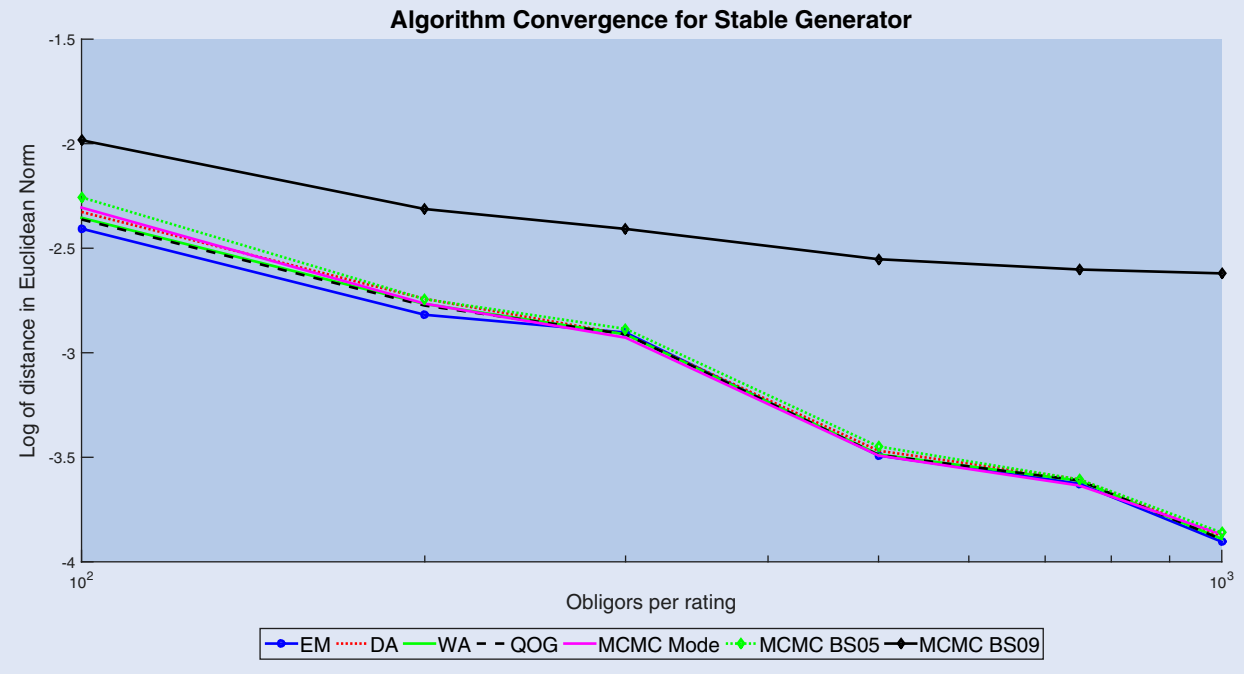

Figure 2. Showing the log of the error for each algorithm as a function of obligors per rating.

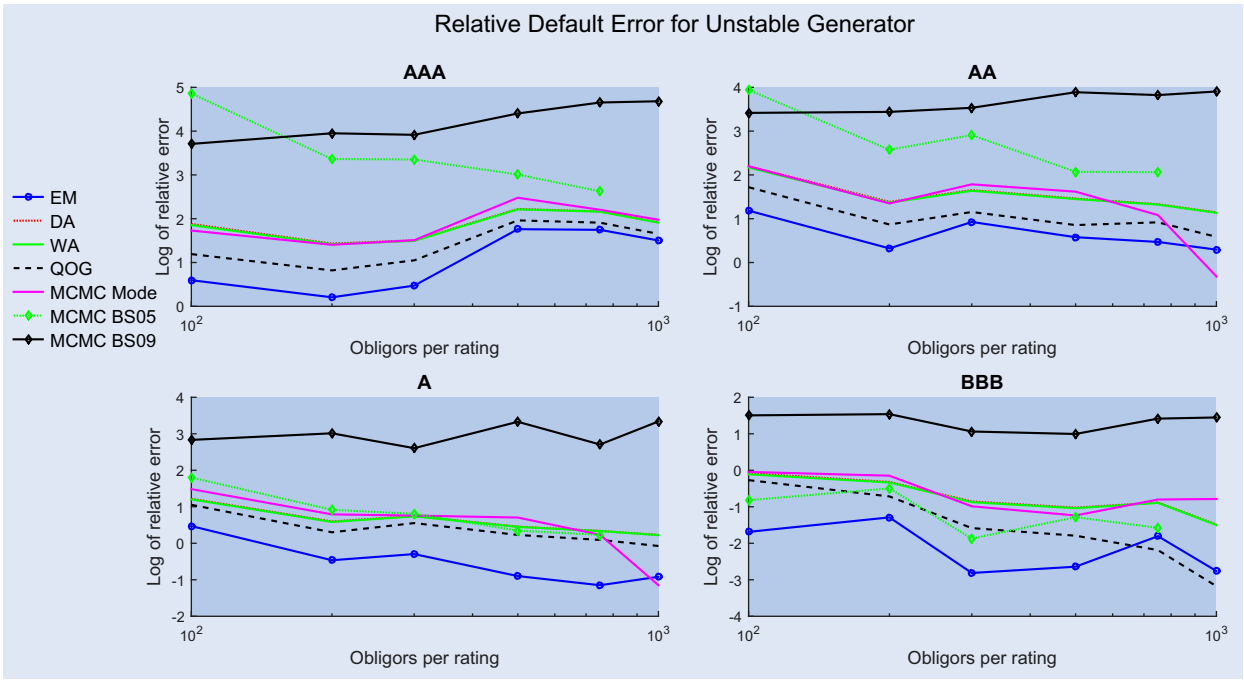

Figure 3. Showing the log of the relative default error for each algorithm as a function of obligors per rating.

same estimated generators as previous we calculate the corresponding one year TPM, that is, we calculate $\exp\{\mathbf{Q}_{\text{estimate}}\}$ (using the `expm` function in MATLAB) for each seed then take the average. The averaged TPM default probabilities are compared to the true ones. To keep the numbers in the comparisons meaningful we plot the log of the relative error, where we define,

$$\text{Relative Error} = \frac{|\text{PD}_{\text{estimate}} - \text{PD}_{\text{true}}|}{\text{PD}_{\text{true}}}.$$

The results of which are given in Figures 3 and 4.

Unlike the overall error, there appears to be far greater volatility in the error estimation w.r.t. the probability of default. Moreover, there appears to be no general downward trend in error for the investment grade ratings. A likely cause for this

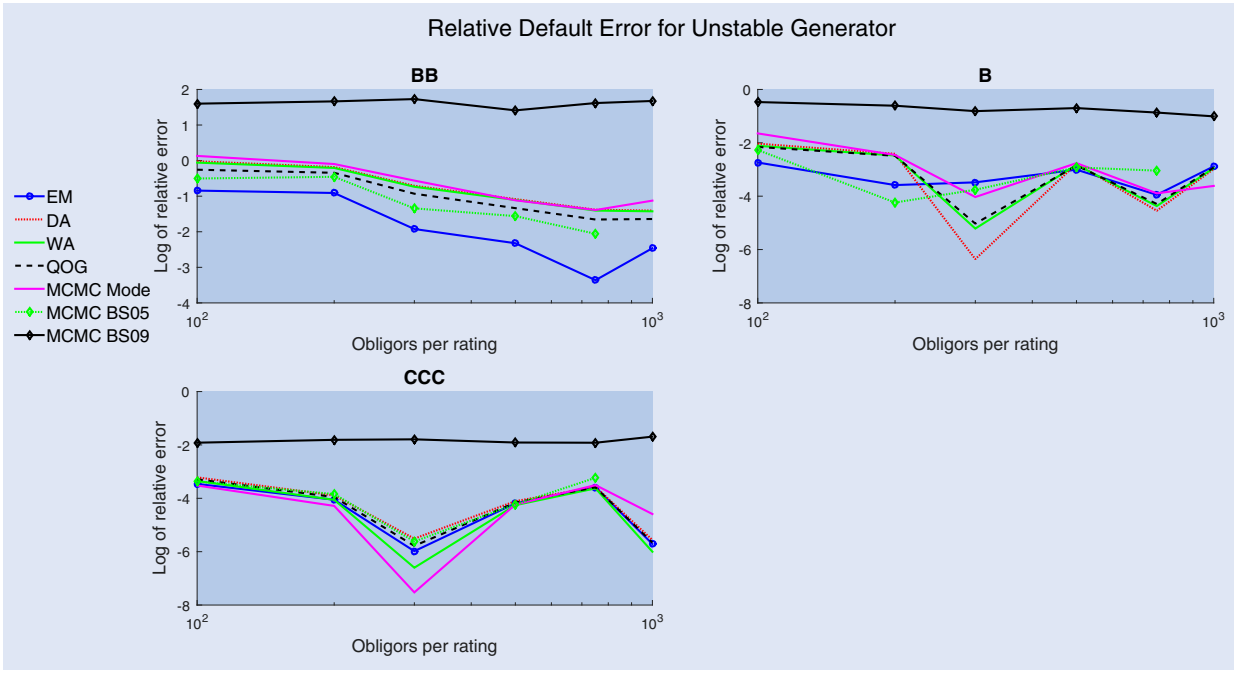

Figure 4. Showing the log of the relative default error for each algorithm as a function of obligors per rating.

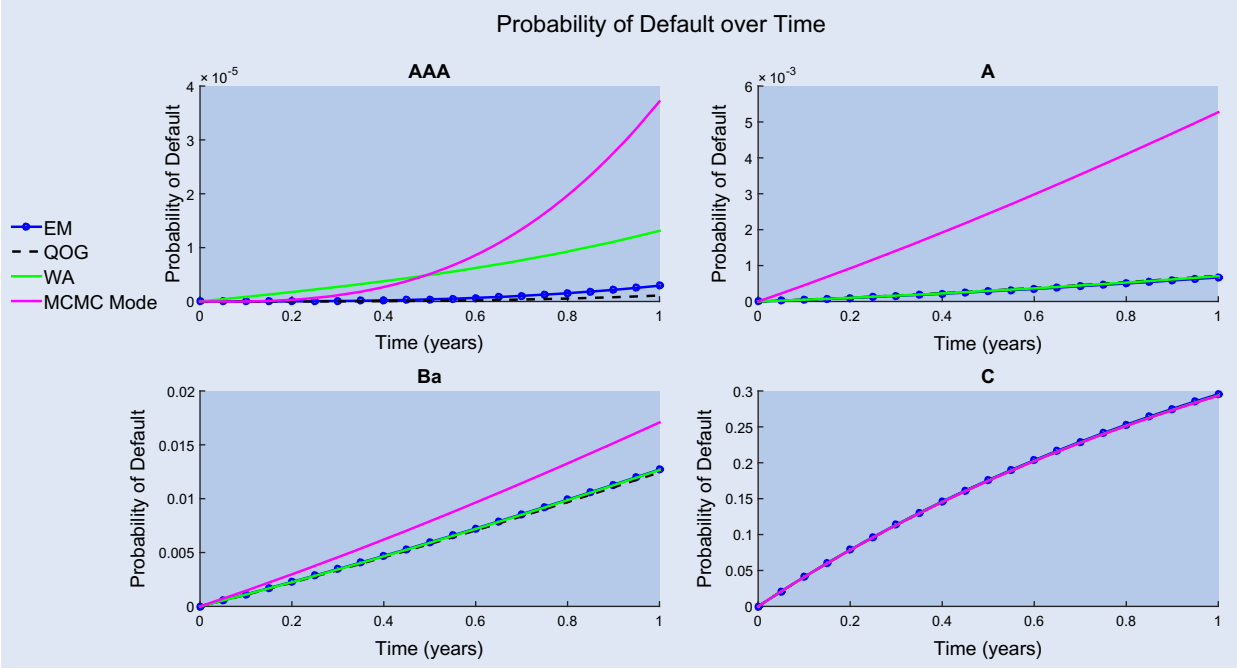

Figure 5. Probability of default over time for EM, QOG, MCMC Mode and WA.

is, even with 1000 companies there are still no/few investment grade defaults. Of the algorithms `MCMC BS09` performs the worst. The EM algorithm though has consistently one of the smallest errors and is clearly the best in the investment grades. We have only shown the results for the unstable generator, the stable generator was similar.

4.2. Time dependent probability of default

A key question that has not been addressed in the literature is how do probabilities of default change in time among the several algorithms. For this, we only consider EM, QOG, WA and the MCMC Mode algorithm from Inamura (2006), since these algorithms gave the best probability of default estimates.

Table 3. Order of time taken to execute the various algorithms. Note that MCMC also depends on the level of information i.e. obligors in each rating. We also note that BS 09 algorithm is faster than the other MCMC algorithms but still takes 10^4 seconds in the case of 1000 obligors per rating.

Algorithms	Deterministic	EM	MCMC
Time (seconds)	< 1	~ 10	$\sim 10^3$ to $\sim 10^4$

We consider a non-embeddable TPM, then estimate the generator matrix $\mathbf{Q}$, from $\mathbf{Q}$ we can easily calculate the probability of a company with some initial rating defaulting in time $t > 0$. The goal here is to assess how that probability changes with time. The TPM is given in table 4, for the MCMC algorithm we took this table to be generated with 250 obligors per rating.

The probability of default across ratings over the one year time horizon is found in Figure 5. The plots give a deeper understanding to the algorithms themselves. As the probability of default increases the algorithms converge, however, in the case of less defaults we observe a much larger discrepancy. This can be thought of as the algorithm's ability to deal with missing data, in the lower grades we observe defaults and thus have a handle on the probability, however, in the case of AAA ratings we observe no defaults, and therefore, it is an approximation by the algorithm. This shows the difference between the methods, shows the potential prior dependence in the MCMC algorithm. What is also extremely interesting is that QOG set the jump in the generator from AA to C as zero (even though the TPM has a non-zero entry there), this implies QOG may in some places under estimate the risk for the investment ratings, this can be seen by the fact QOG puts a smaller probability of default on AAA.

To assess the performance of each algorithm we measure the error by the following,

$$\text{Risk Error} = \frac{\frac{1}{N}\sum_{i=1}^{N} |\text{Risk Charge Estimate}(i) - \text{Risk Charge True}|}{\text{Risk Charge True}},$$

where Risk Charge Estimate(i) is the ith realization of the risk charge and N is the number of TPM sets (10 here). The results obtained by the algorithms are shown in table 9.

There is a clear overestimation of the probability of default at higher grades by the WA and MCMC algorithms.

4.3. Risk charge

The previous tests have been rather theoretical; we now consider a practical test to assess the performance of these algorithms in calculating risk charges. We do not give much discussion to the calculation of these risk charges for more technical details readers should consult texts such as Skoglund and Chen (2011). Here we consider multiple stylized portfolios to represent the risk appetites of different banks. To best of our knowledge analysis into how different risk measures react to different portfolio types has not been considered in the literature. The risk charges we consider are IRC (VaR at 99.9% with a three months liquidity horizon including mark to market loss), IDR (VaR at 99.9% over one year only considering default) and a theoretical risk charge which is IRC but measured using Expected Shortfall (ES) at 97.5%. The final risk charge is included due to the Basel committee showing an increasing interest in ES. We consider four years worth of simulated data, and to keep the analysis realistic we consider 200 companies per rating. We consider three different portfolios corresponding to risk adverse (all investment grade), a speculative portfolio (all speculative grades) and finally a mixed portfolio. The portfolios considered are given in tables 5–7. The tables show the values and ratings of the various bonds in each portfolio.

Alongside these portfolios we calculate the risk charges using the following information,

- The interest rates we receive for a bond in each rating are

AAA	AA	A	BBB	BB	B	C
2.65%	2.69%	2.78%	2.93%	3.18%	5.45%	12.39%

These figures are based on interest rates from Moody's and can be found in Section 4.1 of Skoglund and Chen (2011). Although these interest rates do not technically match the generators we are using for the TPMs they provide reasonable interest rates for our toy example.

- We assume that all money is lost in the case of default (zero recovery rate).
- We calculate credit migration using the one factor† credit metrics model (Gupton *et al.* 1997), i.e. normalized asset returns follow,

$$z_i = \beta_i X + \sqrt{1-\beta_i^2}\epsilon_i\,,$$

where X is the systematic risk, ϵ_i is the idiosyncratic risk both standard normally distributed and β_i is the correlation to the systematic risk, defined in Supervision (2003, p. 50),

$$\beta_i = 0.12\left(\frac{1-\exp\{-50\,P_i^D\}}{1-\exp\{-50\}}\right) + 0.24\left(1-\frac{1-\exp\{-50\,P_i^D\}}{1-\exp\{-50\}}\right),$$

where P_i^D is the probability of default of asset i. Consequently we see that the higher P_i^D the lower the value of β.
- Although more sophisticated methods are available for calculation of VaR and ES (see Fermanian 2014), we calculate the risk charges using Monte Carlo. This is sufficient here since the portfolios are small relative to a typical bank portfolio, therefore we can obtain accurate estimates using a reasonable number of simulations.
- Again, we calculate 10 realizations of the TPMs and estimate a generator for each.

We consider 15×10^5 simulations for each portfolio, to assess whether this was sufficient we calculated VaR and ES using 7.5×10^5, 10×10^5, 12.5×10^5 and 15×10^5 simulations and found the difference between 7.5×10^5 and 15×10^5 to be

†This is technically not the true regulation for the calculation of IDR which requires a two factor model, however our goal here is only to use these calculations as a method for comparing algorithms.

Table 4. Observed TPM used to estimate the generators in probability of default plots.

	AAA	AA	A	BBB	BB	B	C	D
AAA	0.8824	0.1176	0	0	0	0	0	0
AA	0.0064	0.9111	0.0813	0.0008	0.0001	0	0.0003	0
A	0.0003	0.0559	0.8836	0.0499	0.0079	0.0015	0.0002	0.0007
BBB	0	0.0116	0.1585	0.7640	0.0528	0.0070	0	0.0061
BB	0	0	0.0213	0.1193	0.7746	0.0623	0.0099	0.0127
B	0	0	0.0062	0.0199	0.1669	0.7017	0.0730	0.0322
C	0	0	0	0	0.0417	0.2083	0.4544	0.2956
D	0	0	0	0	0	0	0	1

Table 5. Mixed portfolio.

AAA	100, 500, 1500, 750
AA	200, 750, 2000, 650
A	150, 400, 400
BBB	300, 500, 150, 1500
BB	500, 250, 700
B	200, 500
C	100, 150, 200

Table 6. Investment portfolio.

AAA	1000, 500, 1500, 1500
AA	100, 400, 750, 2000, 400, 1500
A	150, 100, 800, 400, 200
BBB	
BB	
B	
C	

Table 7. Mixed portfolio.

AAA	
AA	
A	
BBB	
BB	1000, 150, 100, 800, 1500
B	100, 300, 400, 750, 2000, 1500
C	400, 500, 400, 1000

$< 5\%$ for all cases. Hence were are confident that 15×10^5 gives sufficiently accurate results for our purposes.

With respect to the risk charge calculation, similar to the previous analysis, we calculate the risk charges for every set of TPMs, then average over all the seeds to obtain the risk charge. The risk charges as set by the true generators are given in table 8.

It should be noted, in the stable IDR some algorithms produce a non-zero value for the investment portfolio, therefore, we have inserted the money value. The first observation we make is, all algorithms overestimate the risk for the investment portfolio. This is down to two key feature, one is the 'step like' nature of VaR, where in a small portfolio, small probability changes can make a large difference. The other is because we are averaging over multiple Monte Carlo simulations, thus having one default in one of those realizations will change the overall average dramatically. In terms of a typical bank portfolio this type of error should not be a problem since we would be dealing with a far larger number of assets and hence one would obtain multiple defaults. However, the results do still give a useful comparison between the algorithms. Although the MCMC algorithms can outperform the deterministic algorithms for the speculative grades, remarkably in all categories the EM produces the best results. From the tests we have considered we conclude the EM to be the superior algorithm for this problem.

4.4. Error estimation of the EM algorithm

A major advantage of the statistical algorithms over their deterministic counterparts is that one can derive error estimates (confidence intervals) without the brute force (slightly ad-hoc) method of bootstrapping. For MCMC this comes by looking at the posterior distribution, which we get for free. However, as we have seen MCMC is computationally expensive and we have derived a relatively cheap formula to calculate the confidence intervals from the EM algorithm.

In a similar fashion to the analysis we have carried out previously we now test the error estimate given by the EM. Again, we mask the true generator using simulated TPMs, however, here we only consider the scenario of 300 obligors per rating, but the number of years worth of data is varied. That is, we simulate 50 years worth of TPMs and then apply the EM algorithm using 1 year worth then 2 years etc up to 50 years. This analysis shows both the estimated error for the parameter and also how the error changes when more information is added. It should also be noted that we replace companies who have defaulted to the rating they were pre default. This keeps the number of companies in the system constant and can be thought of as the flow of new companies being rated. Moreover, this is only one realization of the data, hence the parameter estimate and confidence intervals are not particularly smooth.

The transitions shown in Figures 6 and 7 were chosen to show a spectrum of the magnitudes in the generators, the other entries not shown are similar. The first point to make is that the true value of the parameter is almost always within the confidence interval and the confidence interval shrinks as the number of years increases. One of most important features

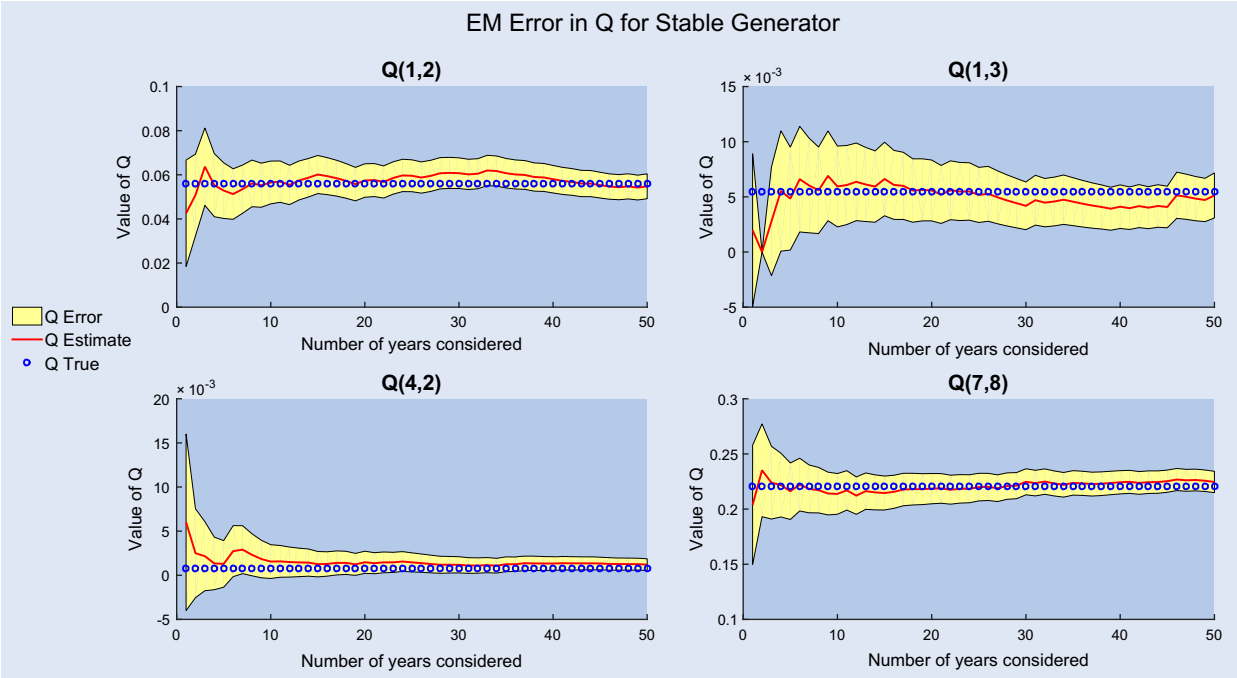

Figure 6. Showing the estimated 95% confidence interval for parameters as a function of years.

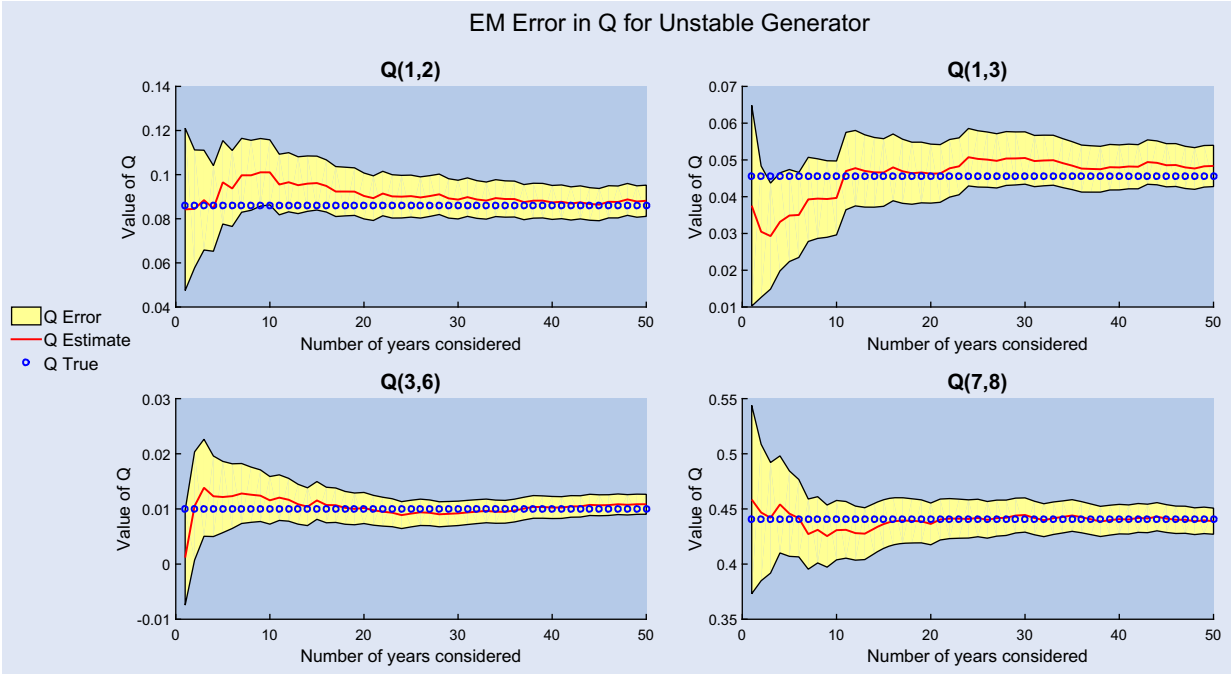

Figure 7. Showing the estimated 95% confidence interval for parameters as a function of years.

Table 8. Risk charge results for the true generators.

	Stable			Unstable		
	Mixed	Investment	Speculative	Mixed	Investment	Speculative
IRC	£702	£0.32	£3395	£1251	£0.41	£5057
IRC ES	£508	£0.20	£2409	£842	£3.78	£3826
IDR	£750	£0	£3400	£1750	£200	£4600

Table 9. Risk charge results for each algorithm as a %.

		Stable			Unstable		
		Mixed	Investment	Speculative	Mixed	Investment	Speculative
IRC	EM	7.3	7.5	1.5	22.5	29 195	2.6
	DA	11.9	8.1	2.4	36.9	66 829	4.3
	WA	11.8	8.1	2.3	37.3	69 293	4.1
	QOG	11.6	7.8	2.3	26.7	38 976	4.1
	MCMCBS05	154	306 000	2	49.6	478 000	4.1
	MCMCBS09	24.9	18.4	14.4	68.3	264 000	14
	MCMCMode	12.5	8.1	3.6	34.9	39 000	3.9
IRC ES	EM	5.3	115	3.4	8.6	375	2.7
	DA	8.2	235	5.1	16.6	1130	3.9
	WA	7.8	210	5	16.4	1109	3.8
	QOG	7.3	123	4.9	12.6	622	3.8
	MCMCBS05	35.4	135 000	4.7	19.7	5315	4.1
	MCMCBS09	21	610	15.5	67.7	6693	13.1
	MCMCMode	9.2	235	6.1	19.1	1063	3.5
IDR	EM	6	0	0.3	4.3	113	3.5
	DA	10	0	1.2	8.6	295	5.7
	WA	9.3	0	0.6	8.6	295	5.2
	QOG	7.3	0	0.6	5.4	185	5.3
	MCMCBS05	139	£1580	0.3	12.6	530	4.7
	MCMCBS09	20	£40	9.3	33.7	775	13.2
	MCMCMode	10	£10	0.9	8	278	4.7

though is that the confidence interval is only small when the EM is stable and close to the true parameter, hence the EM is not 'over confident' but is also providing reasonable estimates on its own error. The final point to make is, although some confidence intervals go below zero by a small amount, this is only true in the case where the parameter is extremely close to zero initially and further, once more data is considered, all parameters have confidence interval which are strictly positive.

Remark 4.1 [Confidence intervals in practice and regulation] Figures 6 and 7 show a very important feature. Namely, how much the estimate can vary with data, especially when the parameter is reasonably small. Such analysis exposes the variance in the estimate and how much data are actually required before the estimate levels out. From this example though the confidence intervals calculated from the information matrix appear to be able to capture this error. In the view of future regulation it may be prudent to take such confidence intervals into account when considering risk charges.

4.4.1. Connection to the global maximum. A previous problem with the EM was one could not be sure of the nature of the stationary point. However, we know the form of the Hessian, and therefore, we can easily check if this point is a maximum by assessing the eigenvalues of this matrix. Clearly, if we were not at a maximum, then it would be worth perturbing the outputted generator and rerunning the algorithm. As discussed in Remark 2.11 the question of a global maximum is very difficult in this setting.

Remark 4.2 One way that has been suggested to improve the chances of the EM converging to the global maximum is, to start from multiple points. Here we can consider creating starting points by setting for each $i \neq j$, $q_{ij} \sim \text{Exp}(\lambda)$ for an appropriate λ then setting q_{ii} appropriately.

We tested the EM according to the above remark and found in every case considered the EM always returns the same generator.

5. Conclusions and future research

In this manuscript, we built upon the closed form expressions for the expected number of jumps and holding times of a CTMC with an absorbing state, over given observations and used the results to derive a closed form expression for the Hessian of the likelihood. This coupled with stronger convergence has elevated the EM algorithm to be the optimal algorithm to tackle this problem.

Across the battery of tests carried out, the EM algorithm outperforms competing algorithms. The EM is a tractable algorithm, slower than the deterministic algorithms but still several orders of magnitude faster than the Markov-Chain Monte-Carlo alternatives (table 2). The statistical algorithms (EM and MCMC) embed a strong robustness property for the estimator contrary to the deterministic algorithms, i.e. the likelihood is far less sensitive to small changes in the underlying TPM. In terms of estimating risk charges, the EM algorithm has superior results in all scenarios.

On the more practical side, Figure 5 highlights that for lower ratings algorithms produce essentially the same estimates for the probabilities of default while a palpable difference emerges at higher ratings. Moreover, the error estimates in the EM may provide a sensible way to test in effect model risk.

Lastly, non-Markovian phenomena like rating momentum (see Lando and Skodeberg 2002) and appropriate models to tackle it will be addressed in forthcoming research.

Acknowledgements

The authors would like to thank Dr. R. P. Jena at Nomura Bank plc London for the helpful comments. In addition, the authors would like to thank Ruth King (U. of Edinburgh), Ioannis Papastathopoulos (U. of Edinburgh) and Samuel Cohen (Oxford Uni.) for the helpful discussions. We thank as well the two anonymous referees for their comments which led to improvements on the initial submission.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

G. Smith was supported by The Maxwell Institute Graduate School in Analysis and its Applications, a Centre for Doctoral Training funded by the UK Engineering and Physical Sciences Research Council (grant [EP/L016508/01]), the Scottish Funding Council, Heriot-Watt University and the University of Edinburgh. G. dos Reis was supported by the *Fundação para a Ciência e a Tecnologia* (Portuguese Foundation for Science and Technology) through the project [UID/MAT/00297/2013] (Centro de Matemática e Aplicações CMA/FCT/UNL).

ORCID

G. dos Reis http://orcid.org/0000-0002-4993-2672

References

Bangia, A., Diebold, F.X., Kronimus, A., Schagen, C. and Schuermann, T., Ratings migration and the business cycle, with application to credit portfolio stress testing. *J. Bank. Financ*, 2002, **26**, 445–474.

Besag, J. and Green, P.J., Spatial statistics and Bayesian computation. *J. R. Stat. Soc. Ser. B (Method.)*, 1993, **55**, 25–37.

Bladt, M., Meini, B., Neuts, M.F. and Sericola, B., Distributions of reward functions on continuous-time Markov chains. In *Matrix-analytic Methods*, edited by G. Latouche, pp. 39–62, 2002 (World Scientific: River Edge, NJ).

Bladt, M. and Sorensen, M., Statistical inference for discretely observed Markov jump processes. *J. R. Stat. Soc. Ser. B (Stat. Method.)*, 2005, **67**, 395–410.

Bladt, M. and Sorensen, M., Efficient estimation of transition rates between credit ratings from observations at discrete time points. *Quant. Finance*, 2009, **9**, 147–160.

Brigo, D., Mai, J.F. and Scherer, M.A., Consistent iterated simulation of multi-variate default times: A Markovian indicators characterization. 2014. Available at: SSRN 2274369.

Cantor, R., An introduction to recent research on credit ratings. *J. Bank. Financ.*, 2004, **28**, 2565–2573.

Christensen, J.H., Hansen, E. and Lando, D., Confidence sets for continuous-time rating transition probabilities. *J. Bank. Financ.*, 2004, **28**, 2575–2602.

Cont, R., Deguest, R. and Scandolo, G., Robustness and sensitivity analysis of risk measurement procedures. *Quant. Finance*, 2010, **10**, 593–606.

Culver, W.J., On the existence and uniqueness of the real logarithm of a matrix. *Proc. Amer. Math. Soc.*, 1966, **17**, 1146–1151.

Cuthbert, J.R., The logarithm function for finite-state Markov semigroups. *J. London Math. Soc.*, 1973, **2**, 524–532.

Dehay, D. and Yao, J.F., On likelihood estimation for discretely observed Markov jump processes. *Aust. New Zeal. J. Stat.*, 2007, **49**, 93–107.

Fermanian, J.D., The limits of granularity adjustments. *J. Bank. Financ.*, 2014, **45**, 9–25.

Frydman, H. and Schuermann, T., Credit rating dynamics and Markov mixture models. *J. Bank. Financ.*, 2008, **32**, 1062–1075.

Gelfand, A.E. and Carlin, B.P., Maximum-likelihood estimation for constrained-or missing-data models. *Can. J. Stat.*, 1993, **21**, 303–311.

Gilks, W.R., Richardson, S. and Spiegelhalter, D.J., Introducing Markov Chain Monte Carlo. *Markov Chain Monte Carlo in Practice*, Vol. 1, pp. 19, 1996 (Chapman & Hall: London).

Gupton G.M., Finger C.C. and Bhatia M., *Creditmetrics: Technical document*, 1997 (JP Morgan & Co).

Hobolth, A. and Jensen, J.L., Summary statistics for endpoint-conditioned continuous-time Markov chains. *J. Appl. Prob.*, 2011, **48**, 911–924.

Inamura Y., Estimating continuous time transition matrices from discretely observed data. Technical report, Citeseer, (No. 06-E-7). Bank of Japan, 2006.

Israel, R.B., Rosenthal, J.S. and Wei, J.Z., Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings. *Math. Financ.*, 2001, **11**, 245–265.

Jarrow, R.A., Lando, D. and Turnbull, S.M., A Markov model for the term structure of credit risk spreads. *Rev. Financ. Stud.*, 1997, **10**, 481–523.

Korolkiewicz, M.W., A dependent hidden Markov model of credit quality. *Int. J. Stoch. Anal.*, 2012, **2012**.

Kreinin, A. and Sidelnikova, M., Regularization algorithms for transition matrices. *Algo Res. Quart.*, 2001, **4**, 23–40.

Kremer, A. and Weißbach, R., Consistent estimation for discretely observed Markov jump processes with an absorbing state. *Stat. Paper*, 2013, **54**, 993–1007.

Kremer, A. and Weißbach, R., Asymptotic normality for discretely observed Markov jump processes with an absorbing state. *Stat. Prob. Lett.*, 2014, **90**, 136–139.

Küchler, U. and Sorensen, M., *Exponential Families of Stochastic Processes*, Vol. 3, 1997 (Springer-Verlag: New York).

Lando, D. and Skodeberg, T.M., Analyzing rating transitions and rating drift with continuous observations. *J. Bank. Financ.*, 2002, **26**, 423–444.

Lin, L., Roots of stochastic matrices and fractional matrix powers. PhD Thesis, University of Manchester, 2001.

Little, R.J.A. and Rubin, D.B., *Statistical Analysis with Missing Data*, 2002 (John Wiley & Sons: Hoboken, NJ).

Long, K., Keenan, S.C., Neagu, R., Ellis, J.A. and Black, J.W., The computation of optimised credit transition matrices. *J. Risk Manage. Financ. Inst.*, 2011, **4**, 370–391.

McLachlan, G. and Krishnan, T., *The EM Algorithm and Extensions*, Vol. 382, 2007 (John Wiley & Sons: Hoboken, NJ).

Norris, J.R., *Markov Chains*, Vol. 2, 1998 (Cambridge University Press: Cambridge).

Oakes, D., Direct calculation of the information matrix via the EM. *J. R. Stat. Soc. Ser. B (Stat. Method.)*, 1999, **61**, 479–482.

Pfeuffer, M., ctmcd: An R package for estimating the parameters of a continuous-time markov chain from discrete-time data. *The R J*, 2017, forthcoming.

Pfeuffer, M., Moestel, L. and Fischer, M., An extended likelihood framework for modeling discretely observed credit rating transitions. Technical report, University of Erlangen-Nuremberg, 2017.

Rutkowski, M. and Tarca, S., Regulatory capital modeling for credit risk. *Int. J. Theor. Appl. Financ.*, 2015, **18**, 1550034.

Skoglund, J. and Chen, W., On the choice of liquidity horizon for incremental risk charges: are the incentives of banks and regulators aligned? *J. Risk Model Validation*, 2011, **5**, 37–57.

Supervision B.C.o.B., *The New Basel Capital Accord*, 2003.

Supervision B.C.o.B., *Fundamental Review of the Trading Book: A Revised Market Risk Framework*, 2013.
Tanner, M.A. and Wong, W.H., The calculation of posterior distributions by data augmentation. *J. Amer. Stat. Assoc.*, 1987, **82**, 528–540.
Trück, S. and Özturkmen, E., Estimation, adjustment and application of transition matrices in credit risk models. In *Handbook of Computational and Numerical Methods in Finance*, edited by Svetlozar T. Rachev, pp. 373–402, 2004 (Birkhäuser: Boston, MA).
Tsai, H. and Chan, K., A note on parameter differentiation of matrix exponentials, with applications to continuous-time modelling. *Bernoulli*, 2003, **9**, 895–919.
Van Loan, C., Computing integrals involving the matrix exponential. *IEEE Trans. Autom. Contr.*, 1978, **23**, 395–404.
Wilcox, R., Exponential operators and parameter differentiation in quantum physics. *J. Math. Phys.*, 1967, **8**, 962–982.
Wu, C.J., On the convergence properties of the EM algorithm. *Ann. Stat.*, 1983, **11**, 95–103.
Yavin, T., Wang, E., Zhang, H. and Clayton, M.A., Transition probability matrix methodology for incremental risk charge. *J. Financ. Eng.*, 2014, **1**, 1450010 [47 pages].

Appendix 1. Proofs

A.1. Proof of Lemma 2.8

We now provide the proof of Lemma 2.8, all terms used have the same definition as they did when the Lemma was stated. Throughout we assume $i \neq h$, thus from from Assumption 2.7 $\mathbb{P}_{\mathbf{Q}}(X(t) = j|X(0) = i) > 0$ for all $j \in \{1, \dots, h\}$ and $t > 0$. The first inequality we prove is the lower bound on the expected number of jumps. Following the assumptions in Lemma 2.8 and time homogeneity we make the observation

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}] \geq P^u_{ij}\mathbb{P}_{\mathbf{Q}}(K_{ij}(t) \geq 1|X(0) = i, X(t) = j)\,.$$

The above inequality holds because we are only considering $X(0) = i, X(t) = j$ and not all possible combinations of start and end states, moreover, $\mathbb{P}_{\mathbf{Q}}(K_{ij} \geq 1|X(0) = i, X(t) = j) \leq \sum_{n=1}^{\infty} n\mathbb{P}_{\mathbf{Q}}(K_{ij} = n|X(0) = i, X(t) = j)$. We further observe,

$$\mathbb{P}_{\mathbf{Q}}(K_{ij} \geq 1|X(0) = i, X(t) = j) \geq \frac{q_{ij}}{-q_{ii}}\,.$$

Thus the lower bound in inequality (2.7) can be easily obtained. We now prove the upper bound on the expected number of jumps. The first observation we make is for all $\nu \in \{1, \dots, h\}$,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|X(0) = i, X(t) = \nu] = \sup_{\mu \in \{1,\dots,h\}} \mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|X(0) = \mu, X(t) = \nu]\,.$$

To see this, let $\mu \neq i$, then denote by τ_i the first time the process enters state i (if $\mathbb{P}_{\mathbf{Q}}(X(t) = i|X(0) = \mu) = 0$ for $t > 0$, then the result is trivial), by the law of total probability we find,

$$\begin{aligned}\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = \mu, X(t) = \nu] &= \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = \mu, X(t) = \nu, \tau_i < t] \\ &\quad \times \mathbb{P}_{\mathbf{Q}}(\tau_i < t|X(0) = \mu, X(t) = \nu) \\ &\quad + \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = \mu, X(t) = \nu, \tau_i \geq t] \\ &\quad \times \mathbb{P}_{\mathbf{Q}}(\tau_i \geq t|X(0) = \mu, X(t) = \nu)\,.\end{aligned}$$

The second term is zero. Then, using the Markov property we obtain,

$$\begin{aligned}\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = \mu, X(t) = \nu] &\leq \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(\tau_i) = i, X(t) = \nu, \tau_i < t] \\ &\leq \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = i, X(t) = \nu]\,.\end{aligned}$$

Consequently from this observation and (2.6) we obtain,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}] \leq hN \sum_{\nu=1}^{h} \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = i, X(t) = \nu]\,.$$

Observe that,

$$\begin{aligned}\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = i, X(t) = \nu] &= \frac{\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)\ \ {}_{\{X(t)=\nu\}}|X(0) = i]}{\mathbb{P}_{\mathbf{Q}}(X(t) = \nu|X(0) = i)} \\ &\leq \frac{\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = i]}{\mathbb{P}_{\mathbf{Q}}(X(t) = \nu|X(0) = i)}\,.\end{aligned}$$

The numerator is easy to bound by considering the expected number of jumps out of i,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|X(0) = i] \leq -q_{ii}t\,.$$

The denominator requires further analysis, firstly, let $n = |i - \nu|$, and therefore by Assumption 2.7 we can go from state i to ν in n jumps, w.l.o.g. let $i \geq \nu$ (it will become clear that the ordering does not matter). Firstly, if $i = \nu$ then,

$$\mathbb{P}_{\mathbf{Q}}(X(t) = \nu|X(0) = i) \geq e^{q_{ii}t}\,.$$

For $i > \nu$, we use the Markov property to obtain,

$$\begin{aligned}&\mathbb{P}_{\mathbf{Q}}(X(t) = \nu|X(0) = i) \\ &\quad \geq \prod_{a=1}^{n} \mathbb{P}_{\mathbf{Q}}\left(X\left(\frac{a}{n}t\right) = i + a \middle| X\left(\frac{a-1}{n}t\right) = i + a - 1\right)\,.\end{aligned}$$

Conditioning on X only making one jump in each increment we obtain,

$$\begin{aligned}&\mathbb{P}_{\mathbf{Q}}(X(t) = \nu|X(0) = i) \\ &\quad \geq \prod_{a=1}^{n} \frac{q_{i+a-1,i+a}}{-q_{i+a-1,i+a-1}}(-q_{i+a-1,i+a-1})t\exp\{q_{i+a-1,i+a-1}t\} \\ &\quad \geq \prod_{a=1}^{n} \epsilon t \exp\{-ht/\epsilon\}\,.\end{aligned}$$

As $n \leq h$ and the terms are strictly smaller than 1, the sought result follows (independent of $\nu \neq i$).

The last inequality to prove concerns the holding times. By taking for $P^u_{ii} > 0$,

$$\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}] \geq P^u_{ii}\mathbb{E}_{\mathbf{Q}}[S_i(t)|X(0) = i, X(t) = i] \geq P^u_{ii}t\exp\{q_{ii}t\}\,,$$

where the final inequality follows by simply considering the case of no jumps. We can then apply the bounds from Assumption 2.7 to complete the inequality.

A.2. Proof of Theorem 2.14

We recall from Wilcox (1967); Tsai and Chan (2003) that for a square matrix $\mathbf{M}$ whose elements depend on a vector of parameters $\{\lambda_1, \dots \lambda_r\}$ (for $r \in \mathbb{N}$), the following identity holds

$$\frac{\partial e^{\mathbf{M}(\boldsymbol{\lambda})t}}{\partial \lambda_i} = \int_0^t e^{(t-u)\mathbf{M}(\boldsymbol{\lambda})} \frac{\partial \mathbf{M}(\boldsymbol{\lambda})}{\partial \lambda_i} e^{u\mathbf{M}(\boldsymbol{\lambda})}\mathrm{d}u\,, \tag{A1}$$

for all $i \in \{1, \dots, r\}$. Let $\mu, \nu, \alpha, \beta \in \{1, \dots, h\}$. Recalling Proposition 2.4, differentiating $\mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y]$ w.r.t. $q_{\alpha\beta}$ yields,

$$\begin{aligned}&\frac{\partial}{\partial q_{\alpha\beta}}\mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y] \\ &\quad = \sum_{s=1}^{n-1} -(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s,y_{s+1}} \left(\frac{\partial}{\partial q_{\alpha\beta}} e^{\mathbf{Q}(t_{s+1}-t_s)}\right)_{y_s,y_{s+1}} \\ &\quad \quad \times (e^{\mathbf{C}^{(\mu\nu)}_{\gamma}(t_{s+1}-t_s)})_{y_s,h+y_{s+1}} + (e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s,y_{s+1}} \\ &\quad \quad \times \left(\frac{\partial}{\partial q_{\alpha\beta}} e^{\mathbf{C}^{(\mu\nu)}_{\gamma}(t_{s+1}-t_s)}\right)_{y_s,h+y_{s+1}}\,.\end{aligned}$$

Note that although the expected value of K only depends on individual elements of the matrix and not the full matrix, we are still able to use the differentiation result since $A_{ij} = \mathbf{e}_i^{\mathcal{T}}\mathbf{A}\mathbf{e}_j$. Hence, from (A1) we obtain,

$$\begin{aligned}
&\frac{\partial}{\partial q_{\alpha\beta}}\mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y]\\
&\quad=\sum_{s=1}^{n-1}-(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s,y_{s+1}}\left(\int_0^t e^{(t-u)\mathbf{Q}}\frac{\partial\mathbf{Q}}{\partial q_{\alpha\beta}}e^{u\mathbf{Q}}\mathrm{d}u\right)_{y_s,y_{s+1}}\\
&\quad\times(e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s,h+y_{s+1}}+(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s,y_{s+1}}\\
&\quad\times\left(\int_0^t e^{(t-u)\mathbf{C}_\gamma^{(\mu\nu)}}\frac{\partial\mathbf{C}_\gamma^{(\mu\nu)}}{\partial q_{\alpha\beta}}e^{u\mathbf{C}_\gamma^{(\mu\nu)}}\mathrm{d}u\right)_{y_s,h+y_{s+1}}.
\end{aligned}$$

Clearly, since $q_{\alpha\beta}$ appears twice in $\mathbf{Q}$,

$$\frac{\partial\mathbf{Q}}{\partial q_{\alpha\beta}}=\mathbf{e}_\alpha\mathbf{e}_\beta^{\mathcal{T}}-\mathbf{e}_\alpha\mathbf{e}_\alpha^{\mathcal{T}},\qquad\text{and}$$
$$\frac{\partial\mathbf{C}_\gamma^{(\mu\nu)}}{\partial q_{\alpha\beta}}=\begin{bmatrix}\mathbf{e}_\alpha\mathbf{e}_\beta^{\mathcal{T}}-\mathbf{e}_\alpha\mathbf{e}_\alpha^{\mathcal{T}} & \mathbf{e}_\mu\mathbf{e}_\nu^{\mathcal{T}}\delta_{\mu\alpha}\delta_{\nu\beta}\\ 0 & \mathbf{e}_\alpha\mathbf{e}_\beta^{\mathcal{T}}-\mathbf{e}_\alpha\mathbf{e}_\alpha^{\mathcal{T}}\end{bmatrix}.$$

Then, by Van Loan (1978)

we can solve these integrals explicitly to obtain,

$$\begin{aligned}
&\frac{\partial}{\partial q_{\alpha\beta}}\mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y]\\
&\quad=\sum_{s=1}^{n-1}-(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s,y_{s+1}}\left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)}\right)_{y_s,h+y_{s+1}}\\
&\quad\times(e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s,h+y_{s+1}}+(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s,y_{s+1}}\\
&\quad\times\left(e^{\mathbf{C}_\psi^{(\alpha\beta,\mu\nu)}(t_{s+1}-t_s)}\right)_{y_s,3h+y_{s+1}},
\end{aligned}$$

again $\mathbf{C}_\eta^{(\alpha\beta)}$ and $\mathbf{C}_\psi^{(\alpha\beta,\mu\nu)}$ are as defined in the Theorem's statement.

Therefore, we have a closed form expression for the derivative of expected jumps w.r.t. $q_{\alpha\beta}$. Applying a similar argument for the expected holding time we obtain,

$$\begin{aligned}
&\frac{\partial}{\partial q_{\alpha\beta}}\mathbb{E}_{\mathbf{Q}}[S_\mu(t)|y]\\
&\quad=\sum_{s=1}^{n-1}-(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-2}_{y_s,y_{s+1}}\left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)}\right)_{y_s,h+y_{s+1}}\\
&\quad\times(e^{\mathbf{C}_\phi^{(\mu)}(t_{s+1}-t_s)})_{y_s,h+y_{s+1}}+(e^{\mathbf{Q}(t_{s+1}-t_s)})^{-1}_{y_s,y_{s+1}}\\
&\quad\times\left(e^{\mathbf{C}_\omega^{(\alpha\beta,\mu)}(t_{s+1}-t_s)}\right)_{y_s,3h+y_{s+1}},
\end{aligned}$$

where $\mathbf{C}_\omega^{(\alpha\beta,\mu)}$ is as defined in the Theorem. Combining these yields the required result.

Appendix 2. Overview of Markov Chain Monte Carlo algorithm

For details on the Markov Chain Monte Carlo (MCMC) theory we refer the reader to Gilks *et al.* (1996). Algorithms for implementing MCMC to estimate a generator, from discrete observations are discussed in Bladt and Sorensen (2005) and Bladt and Sorensen (2009). MCMC differs from the EM in the sense that EM estimates the set of parameters which maximizes the likelihood function, while MCMC samples from the posterior distribution. Namely, given some data D, the posterior distribution of parameters θ is $\pi(\theta|D)$, which by Bayes' theorem is,

$$\pi(\theta|D)=\frac{\pi(D|\theta)\pi(\theta)}{\int\pi(D|\theta)\pi(\theta)\mathrm{d}\theta},$$

with $\pi(D|\theta)$ denoting the likelihood and $\pi(\theta)$ the prior distribution. MCMC obtains the best guess of θ by sampling from $\pi(\theta|D)$ and taking the Monte Carlo approximation of the expected value. The reason the expectation is our best guess is due to the fact we use both the data (likelihood) but also our experience on what the outcome should approximately be (the prior). Although the prior can be extremely useful in stopping 'bad' answers it is also a criticism of MCMC due to so-called prior sensitivity.

Remark 2.1 Here we purely discuss MCMC to sample from the posterior, algorithms which approximate the maximum likelihood in the presence of missing data do exist, but are more useful when, for example, one cannot explicitly write the E step in the EM algorithm (see Gelfand and Carlin 1993).

Similar to the case of the EM algorithm the problem faced here is missing data. Namely we wish to consider the so-called posterior distribution of the generator matrix $\mathbf{Q}$, which we denote by $\pi(\mathbf{Q}|D)$ (although it is common to suppress the data and only write $\pi(\mathbf{Q})$). The difficulty is, in its current state this is an extremely hard distribution to evaluate so we augment with an auxiliary variable X (see Gilks *et al.* 1996, p. 105 and Besag and Green 1993). In general X need not require an interpretation, although here it will correspond to the full Markov chain. In order to generate realizations of $\pi(\mathbf{Q}|D)$, we specify the conditional distribution $\pi(X|\mathbf{Q}, D)$ which provides the joint distribution $\pi(\mathbf{Q}, X|D) = \pi(\mathbf{Q}|D)\pi(X|\mathbf{Q}, D)$ and therefore the marginal distribution of $\mathbf{Q}$ is $\pi(\mathbf{Q}|D)$. One can then sample from the marginal distribution using any sampling method that preserves the joint distribution $\pi(\mathbf{Q}, X|D)$ (and by extension $\pi(\mathbf{Q}|D)$), such as Gibbs or Metropolis Hastings.

The method used in Bladt and Sorensen (2005) and Bladt and Sorensen (2009) is the data augmentation algorithm from Tanner and Wong (1987) (see also Little and Rubin 2002, p. 200). We specify the prior distribution $\pi(\mathbf{Q})$ and take a realization from this distribution, $\mathbf{Q}^{(0)}$, we then construct a sequence $\{\mathbf{Q}^{(k)}, X^{(k)}\}$, for $k = 1, \ldots, M$ by:

(i) Draw, $X^{(k)} \sim \pi(X|\mathbf{Q}^{(k-1)}, D)$.
(ii) Draw, $\mathbf{Q}^{(k)} \sim \pi(\mathbf{Q}|X^{(k)}, D) = \pi(\mathbf{Q}|X^{(k)})$ (since $X^{(k)}$ is richer than D).
(iii) Save $\{\mathbf{Q}^{(k)}, X^{(k)}\}$ and take $k = k + 1$.

Under mild conditions (see Gilks *et al.* 1996, Chapter 4), after some burn-in n, the sequence $\{\mathbf{Q}^{(k)}, X^{(k)}\}$ for $k \geq n$ has the same distribution as $\pi(\mathbf{Q}, X|D)$. Moreover, the marginals also have the correct distribution, namely, $\{\mathbf{Q}^{(k)}\} \sim \pi(\mathbf{Q}|D)$ for $k \geq n$. Therefore we estimate the generator matrix by, $\frac{1}{M-n+1}\sum_{k=n}^{M}\mathbf{Q}^{(k)}$.

For the choice of prior, $\pi(\mathbf{Q})$, Bladt and Sorensen (2005) suggest a prior from the gamma distribution with shape α_{ij} and scale $1/\beta_i$. Hence, $q_{ij} \sim \Gamma(\alpha_{ij}, 1/\beta_i)$, where $\alpha_{ij}, \beta_i \geq 0, \forall\, i \neq j \in \{1, \ldots, h\}$. With this choice, the prior is a

conjugate prior. Although this prior has some drawbacks, we note, by assuming the prior to follow a Gamma distribution we effectively bound the parameter space, therefore, there is no need to make the space compact. Noting that the posterior distribution of X is equivalent to the likelihood i.e. $\pi(X|\mathbf{Q}) = L_t(X; \mathbf{Q})$, one has

$$\pi(\mathbf{Q}|X, D) = \pi(\mathbf{Q}|X) = \frac{\pi(\mathbf{Q}, X)}{\pi(X)} \propto L_t(X; \mathbf{Q})\pi(\mathbf{Q}).$$

From the likelihood of a CTMC and the assumption on the prior we infer that,

$$\begin{aligned}
L_t(X; \mathbf{Q})\pi(\mathbf{Q}) &\propto \prod_{i=1}^{h}\prod_{j\neq i} q_{ij}^{K_{ij}(t)}e^{-S_i(t)q_{ij}}\prod_{i=1}^{h}\prod_{j\neq i} q_{ij}^{\alpha_{ij}-1}e^{-\beta_i q_{ij}}\\
&= \prod_{i=1}^{h}\prod_{j\neq i} q_{ij}^{K_{ij}(t)+\alpha_{ij}-1}e^{-(S_i(t)+\beta_i)q_{ij}}.
\end{aligned}$$

We do not have equality here since there is no normalization term. We generate q_{ij} with $i \neq j$ from the distribution $\Gamma\big(K_{ij}(t)+\alpha_{ij}, 1/(S_i(t) + \beta_i)\big)$ (since each q_{ij} is independent).